Phase switch Monte Carlo

McNeil-Watson, Graham

Award date:
2007

Awarding institution:
University of Bath

PHASE SWITCH MONTE CARLO

Submitted by Graham McNeil-Watson
for the degree of
Doctor of Philosophy
of the University of Bath
Department of Physics
October 2007

ProQuest LLC
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106-1346

Abstract

Computational studies of phase behaviour have always proved difficult, since phase transitions are inherently slow processes compared to accessible simulation timescales. Despite valiant efforts by researchers there remains a dearth of efficient, robust and scalable methods for determining phase equilibria, especially in the case of fluid–crystalline solid transitions. This thesis is about such phase coexistence problems, the existing solutions, and more advanced methods that have only recently come into their own.

Extended sampling methods are examined in detail, and applied to a testbed system, the critical point Lennard-Jones fluid, leading to an estimate of the system free energy in the thermodynamic limit. Then a comparatively new technique, phase switch Monte Carlo (*Phys. Rev. Lett.* **85**, 5138) is applied initially to the venerable hard sphere system. The method overcomes many of the shortcomings present in other works by directly connecting the coexisting phases in a single simulation, and doing so without creating an artificial inter-phase route but rather affecting a direct 'phase leap' from one phase to the other. Finally, phase switch is generalised to soft potentials and applied to the Lennard-Jones freezing transition, resulting in an extensive mapping of the phase boundary for a variety of system sizes (*J. Chem. Phys* **124**, 064504).

Acknowledgements

First of all I would like to thank my supervisor, Dr Nigel Wilding, without whom this thesis would have been impossible. Thank you for your guidance, insight, inspiration and no small amount of patience.

I would like to thank Prof Alistair Bruce, Dr Arjun Acharya and Ethan Mastny for valuable discussions.

Thanks also go to my friends, parents, family, colleagues and the department. I could not have done it without you.

Contents

List of Figures

Conventions

$\sum_{<ij>}^{N}$	Sum over all possible unordered pairs of the i and j terms, excluding the $i = j$ case.
$\mathcal{O}(\cdot)$	Indicates scaling characteristics of an algorithm within a multiplicative constant. For example: the calculation of the total energy of a system having a pairwise additive potential is $\mathcal{O}(N^2)$ in the number of particles, N. Only the fastest growing term is included.
1.23(45)	The numbers in parenthesis are interpreted as an error bar such that 1.23(45) becomes 1.23 ± 0.45.

Abbreviations

MC	Monte Carlo
MD	Molecular Dynamics
FCC	Face centred cubic
HCP	Hexagonal close packed
CS	Crystalline solid
F	Fluid
CPU	Central processing unit
CP	Critical point
TP	Triple point
VS	Visited states (algorithm)
WL	Wang-Landau (algorithm)
TM	Transition matrix (algorithm)

Chapter 1

Background

This thesis is about Monte Carlo techniques for determining phase coexistence, in particular: phase switch Monte Carlo. The first three chapters will provide introductory and review material forming the necessary background for the computational studies in the last three chapters. They also create a theoretical framework within which the later techniques can be presented.

1.1 Introduction

Statistical mechanics has been of central importance to our understanding of the physical world. It relates our knowledge of the microscopic interactions between the constituents of matter to the macroscopic properties which that matter manifests. In doing so, it exposes a fundamental principle of the behaviour of such systems: although the macroscopic properties do depend sensitively on the form of the underlying interactions, they do not depend on what any particular molecule is doing. Rather, that bulk properties are effectively averages over a very large number of particles renders the microscopic statistical fluctuations unimportant. Hence a fluid whose molecules are in constant random (strictly: chaotic) molecular motion can still exert a constant pressure, at least to an excellent approximation.

Computer simulation has greatly expanded our ability to study phase transitions. While computational power is growing exponentially, many advances have also been made in simulation methodology. This means that the scale and complexity of systems that can be studied is a constantly moving target. This thesis is about some of these techniques.

The focus of this work will be phase behaviour, in particular the freezing transition, and will look at the long timescale behaviour rather than the details of the dynamics of the transitions themselves. Calculating freezing properties has obvious relevance in materials research.

1.2 Phase behaviour

We will sometimes refer to the particles in our systems as atoms or molecules, but acknowledge here that this is implicitly doing a disservice to the breadth of application of statistical mechanics. The same principles can be used to study the behaviour of: colloidal suspensions, polymer melts, intra-molecular systems, and materials at the magnetic domains level, to name a few.

Stating that a system is in a particular **phase** is synonymous with identifying the general behaviour of the system under the current external conditions. The properties that signify particular phases vary all the way from the microscopic to the macroscopic domains. To formalise this we identify 'external conditions' as **control parameters**, macroscopic properties that are imposed upon the system. Common examples are temperature, pressure, and applied magnetic field strength. As we vary these control parameters the properties of the material will change.

If the phase changes in response to an alteration to the control parameters then the system has undergone a **phase transition**, the most familiar of which are changes of state such as freezing or sublimation, but also include structural changes within a single state of matter. Water, for example, has particularly rich phase behaviour with (at least) two liquid phases [1] and a multitude of solid phases [2].

We characterise phase transitions by identifying one or more observables as **order parameters** for the transition. For the most common example of a **first order phase transition** [3] a useful order parameter will be discontinuous at the particular values of the control parameters corresponding to the transition. Typical order parameters are: density, energy, heat capacity or magnetic susceptibility, but many others are used.

First order phase transitions are characterised by latent heat – the system must absorb or release an amount of energy to make the transition. They are also associated with mixed-phase configurations of the system, in which regions coexisting pure phases are separated by an interface. Similarly we can introduce a **second order phase transition** where there is no latent heat. Values of the control parameters for which a second order phase transition occurs are known as **critical points** and their special

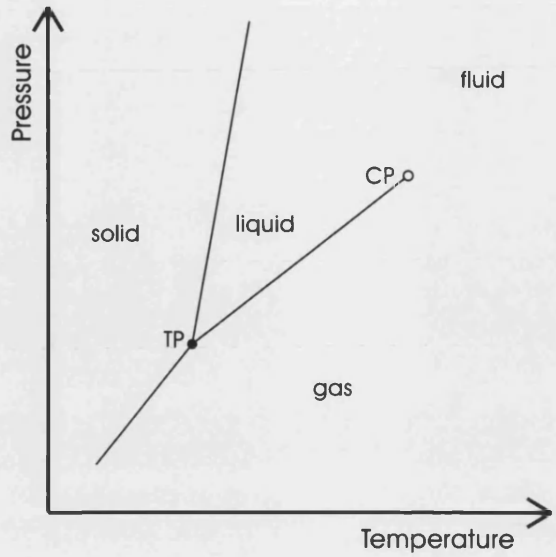


Figure 1.1: Schematic phase diagram from a simple fluid.

significance and so-called **critical behaviour** has been extensively studied [4, 5].

The phase behaviour of a system can be conveniently summarised in a **phase diagram**, which shows which phase is stable at various combinations of the control parameters. Figure 1.1 shows a schematic phase diagram for a simple fluid. At some values of the control parameters two (or more) phases may be stable and we identify **phase coexistence**. This typically occurs for a range of values of the control parameters so creating **lines of coexistence** or a **phase boundary**. Terminology varies; the coexistence line separating fluid and solid, for example, is variously known as: the solid–fluid coexistence line, the melting curve and the freezing line. The point on the phase diagram where liquid, solid and vapour phases all coexist is known as the **triple point**. The liquid–vapour curve terminates in a critical point.

1.3 Classical mechanics

This thesis will consider classical mechanics exclusively; quantum effects will not be treated. Although quantum effects are relevant in the atomic domain, and any classical description of atomic and molecular interaction is necessarily incomplete, this does not compromise the usefulness of studies such as this. When applied to mesoscopic systems, such as colloidal suspensions, we *can* fully justify ignoring quantum effects. Where we

do apply classical mechanics to molecular systems we use the following justifications:

- First of all, any quantum mechanical calculation (such as the ubiquitous Density Functional Theory) is orders of magnitude more computationally expensive and typically has worse computational scaling behaviour than a similar classical calculation, severely limiting the number of atoms that can be simulated. By contrast, useful data can be obtained from classical systems many times larger. Quantum Monte Carlo is orders of magnitude more complicated still [7].
- Secondly, it is well known that macroscopic objects do not demonstrate quantum effects on macroscopic length scales, except in extremely unusual circumstances (vortex formation in Helium-3, Bose-Einstein condensation, etc.). That quantum effects approach a classical limit for large systems further suggests the fundamental utility of classical treatments for studying bulk behaviour.
- Thirdly, using a simpler description allows us to understand the process of phase behaviour more fundamentally: we can discover what the minimum complexity of a system's interactions needs to be to yield 'interesting' phase behaviour. Such studies frequently show that very simple models can reproduce complicated phase behaviour. We can use such data to identify the microscopic origins of particular macroscopic behaviours.
- Finally, classical treatments have empirically proven utility.

We will characterise our classical systems by writing down a Hamiltonian $\mathcal{H}(\mathbf{r}^N, \mathbf{p}^N)$ which gives the energy as a function of the positions and momenta of all the particles (we write $\mathbf{r}^N$ for $\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_N$ and analogously for the momenta). The dynamics themselves will be discussed in §1.6, but the framework of statistical mechanics (developed in the next section) depends purely on the form of the Hamiltonian.

1.4 Statistical mechanics and thermodynamics

Phase behaviour was first understood within the framework of thermodynamics, which deals with matter on macroscopic scales. The field of **statistical mechanics** serves to link the macroscopic properties to the microscopic degrees of freedom of the system. Full treatments can be found in standard textbooks [8, 9].

Only very small systems (compared to the thermodynamic limit) can be simulated with contemporary computer hardware, so we need to take care with the boundary conditions. The particular set of control parameters we impose upon the system determines

which thermodynamic **ensemble** we are working in. The **microcanonical** ensemble, for example, refers to systems of specified (fixed) particle number (N), volume (V) and energy (E) and is often abbreviated as the NVE ensemble. Such a system will have the conjugate thermodynamic variables of temperature, pressure and chemical potential fluctuating, and we will measure these during the simulation. All ensembles become equivalent in the thermodynamic limit so we are free to use whichever proves to be convenient.

We will often refer to **microstates** of a system, meaning simply a set of specific values for each of the microscopic degrees of freedom of the system and may include an observable such as volume or particle number depending on the exact ensemble. We will denote the full set as Γ or Υ for brevity. Microstates may be thought of as spanning a high dimensional vector space with each dimension corresponding to a degree of freedom of the system. This is frequently referred to as **phase space**.

Statistical mechanics makes one (and only one) assumption, founded ultimately in empirical verification, namely that of **equal *a priori* probabilities**, which may be summarised in this context as:

> *All microstates are equally likely to occur in an isolated system at equilibrium.*

An isolated system corresponds to the microcanonical ensemble. By considering one subject system embedded in a heat bath provided by a much larger system we commonly derive the **Boltzmann distribution** [10] which gives us the probability distribution for systems in ensembles where the temperature is fixed (by means of the heat bath) and the energy fluctuates. The distribution is:

$$\mathcal{P}(\Gamma) = \frac{1}{Z} e^{-\beta \mathcal{H}(\Gamma)}, \tag{1}$$

where β is the inverse temperature ($\beta \equiv 1/k_B T$) and Z is a normalisation constant known as the **partition function**:

$$Z = \int e^{-\beta \mathcal{H}(\Gamma)} \, \mathrm{d}\Gamma. \tag{2}$$

The integration extends over the entire phase space.

Now we can construct an **ensemble average** for some observable, $\mathcal{A}$, we are interested in:

$$\langle \mathcal{A} \rangle = \frac{1}{Z} \int \mathcal{A}(\Gamma) \, e^{-\beta \mathcal{H}(\Gamma)} \, \mathrm{d}\Gamma \tag{3}$$

The ensemble average represents a mean value, evaluated over a very large number of copies of the system selected according to the Boltzmann distribution. It is useful to form ratios of observables, in which case the partition function will cancel, leaving:

$$R_{\mathcal{AB}} = \frac{\langle \mathcal{A} \rangle}{\langle \mathcal{B} \rangle} = \frac{\int \mathcal{A}(\Gamma)\, e^{-\beta \mathcal{H}(\Gamma)}\, \mathrm{d}\Gamma}{\int \mathcal{B}(\Gamma)\, e^{-\beta \mathcal{H}(\Gamma)}\, \mathrm{d}\Gamma}. \tag{4}$$

When we have two coexisting phases which are separated by a first order transition then the phase space will be divided into a vast majority of microstates which do not contribute significantly to Z, and disjoint regions corresponding to each coexisting phase. If a suitable order parameter is known (or can be constructed) to robustly differentiate the phases, then we can simply evaluate equation 2 constrained to include each phase in turn, and so yield statistical weights for each of the coexisting phases, Z_1, Z_2, etc. Coexistence can be readily determined as the point at which these are equal. In practice, this computation cannot be performed directly and we must resort to indirect methods to make the calculation. Section 1.7 on the Monte Carlo method will discuss these issues in greater depth.

In general it is not possible to determine Z within computer simulation, however Monte Carlo can, with appropriate caveats, determine ensemble averages such as equation 3 and other *ratios* of constrained partition functions such as equation 4. Consider as an example the freezing of a fluid to a crystalline solid, and set $\mathcal{A}(\Gamma) = \delta_{\mathrm{CS}}(\Gamma)$ and $\mathcal{B}(\Gamma) = 1$ i.e. use a function which gives unity for a crystalline solid microstate and zero for a fluid one. The values of the control parameters for which the ratio becomes $1/2$ would then correspond to phase coexistence.

Two phases separated by a first order transition will occupy disjoint regions of phase space. As such, the microstates which have non-negligible statistical weight can be classified as belonging to one phase or the other (at least in the thermodynamic limit). Conceptually, this allows us to separate the contributions to the total statistical weight for each phase:

$$Z = Z_1 + Z_2 + \ldots + Z_n, \tag{5}$$

where Z_i is integral of equation 2 subject to an additional constraint, namely only microstates identified with phase i are included in the summation. In practice selection of an order parameter which can make this classification unequivocally on a microstate by microstate basis can be difficult.

Coexistence between two (or more) phases occurs when they contribute equally to Z, e.g. $Z_1 = Z_2$. It follows that the **line of coexistence** of two phases in the system

phase diagram is the locus of points for which the phases have equal statistical weight. In the thermodynamic limit the phase boundary is sharp, and the stable phase changes abruptly; whereas in the smaller systems amenable to computer simulation there is a more gradual transition: the statistical weight of the incoming phase increases, goes through coexistence itself, and eventually dominates as we pass through the phase boundary.

The theory of thermodynamics predates statistical mechanics, as well as our knowledge of the atom. It provides an alternative vocabulary: the quantity of central importance is the **free energy**, F, which represents the amount of energy available to do work, excluding energy that is in the form of heat. The free energy attains a minimum under equilibrium conditions. The exact form of the free energy depends on the chosen ensemble. For the isobaric–isothermal (NPT) ensemble, this is the Helmholtz free energy:

$$F = U - TS, \tag{6}$$

where U is the internal energy, T is the temperature and S is the entropy. We can relate the thermodynamic and statistical mechanical descriptions of a system through a **bridge equation**. For the NPT ensemble this would be:

$$F = -k_B T \ln Z. \tag{7}$$

This demonstrates the link between the entropy of a system and the number and statistical weight of the accessible microstates. We will use whichever set of terminology is most useful in each context.

1.5 Separation of the energy

There is an important simplification we can make for classical systems: **separation of the energy**. The Hamiltonian for a system of Newtonian particles can be written as the sum of kinetic and configurational terms:

$$\mathcal{H}(\mathbf{r}^N, \mathbf{p}^N) = \sum_{i=1}^{N} \frac{\mathbf{p}_i^2}{2m_i} + V(\mathbf{r}^N), \tag{8}$$

contingent on the potential function being independent of the momenta. Now if we consider the expectation value of an observable $\mathcal{A}$ under this Hamiltonian, we can

transform:

$$\begin{aligned}
\langle \mathcal{A} \rangle &= \prod_{i=1}^{N} \left\{ \iiint_V d\mathbf{r}_i \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} d\mathbf{p}_i \right\} \mathcal{A}(\mathbf{r}^N) \frac{e^{-\beta \mathcal{H}(\Gamma)}}{Z} \\
&= \frac{1}{Z} \left[\prod_{i=1}^{N} \left\{ \iiint_V d\mathbf{r}_i \right\} \mathcal{A}(\mathbf{r}^N) \, e^{-\beta \mathcal{U}(\mathbf{r}^N)} \right] \left[\prod_{i=1}^{N} \left\{ \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} d\mathbf{p}_i \, e^{-\beta \mathbf{p}_i^2 / 2m} \right\} \right] \\
&= \frac{1}{Z} \left[\prod_{i=1}^{N} \left\{ \iiint_V d\mathbf{r}_i \right\} \mathcal{A}(\mathbf{r}^N) \, e^{-\beta \mathcal{U}(\mathbf{r}^N)} \right] \left[\int_{-\infty}^{+\infty} dx \, e^{-\beta x^2 / 2m} \right]^{3N} \\
&= \frac{1}{Z} \left(\frac{2m\pi}{\beta} \right)^{\frac{3N}{2}} \prod_{i=1}^{N} \left\{ \iiint_V d\mathbf{r}_i \right\} \mathcal{A}(\mathbf{r}^N) \, e^{-\beta \mathcal{U}(\mathbf{r}^N)}.
\end{aligned}$$

Since the same factorisation can also be performed on the partition function itself we can cancel the parenthetical factor between each, in so doing introducing a new reduced partition function which does not depend on the kinetic part. This shows equivalence of results obtained with equation 8 and a simplified Hamiltonian:

$$\mathcal{H}'(\mathbf{r}^N) = V(\mathbf{r}^N). \tag{9}$$

An ensemble average of any observable (that is independent of momentum coordinates) will take the same value as the system of equation 8 and that of 9. The only difference is a scale factor, that is absorbed into the partition function in any case. Thus any quadratic kinetic energy term can be safely ignored when writing down system Hamiltonians. We omit kinetic energy terms throughout.

1.6 Molecular dynamics

Computer simulation is widely used to determine the properties of a system. For a handful of simple models it is possible to perform the integrals in equations 2 and 3 analytically. However, for all but the most trivial systems there is no analytical solution either known or possible. Famously, having three bodies in a system is enough to render it chaotic, with no closed solution. Hence we must appeal to a numerical approach in general. There are two main approaches: Monte Carlo and molecular dynamics, which we discuss here. A good introduction can be found in [6].

Molecular dynamics (MD) attempts to approximate the actual behaviour of a real system by integrating the equations of motion directly. This very general approach actually provides much more information than the static equilibrium averages (cf. equation 3); dynamic properties such as diffusion constants, transport coefficients and mean-

free path lengths can all be evaluated directly.

We proceed from Newton's equations:

$$m_i \ddot{\mathbf{r}}_i = -\nabla V(\mathbf{r}_i) \tag{10}$$

where V is the potential induced by all the other particles, and any external field. For a pair-wise additive potential this would be:

$$\nabla V(\mathbf{r}_i) = \sum_{\substack{j=1 \\ j \neq i}}^{N} U'(|\mathbf{r}_i - \mathbf{r}_j|) \tag{11}$$

where U' is the derivative of the potential function with respect to its argument.

To integrate these equations we typically make successive small fixed steps (Δt) in time (t), updating the particle positions ($\mathbf{r}_i(t)$), velocities ($\mathbf{v}_i(t)$) and forces acting on each particle ($f_i(t)$). The most obvious integration scheme is known as **Euler integration**:

$$\begin{aligned} \mathbf{v}_i(t+\Delta t) &\leftarrow \mathbf{v}_i(t) + \frac{f_i(t)}{m_i}\Delta t \\ \mathbf{r}_i(t+\Delta t) &\leftarrow \mathbf{r}_i(t) + \mathbf{v}_i(t+\Delta t)\Delta t, \end{aligned}$$

where t is time. It can be shown that the errors are of order $O(\Delta t^2)$. Many other integration schemes exist with superior error characteristics. We show, for example, the **Verlet** algorithm [12]:

$$\begin{aligned} \mathbf{r}_i(t+\Delta t) &\leftarrow 2\mathbf{r}_i(t) - \mathbf{r}_i(t-\Delta t) + \frac{f_i(t)}{m_i}\Delta t^2 \\ \mathbf{v}_i(t) &\leftarrow \frac{\mathbf{r}_i(t+\Delta t) + \mathbf{r}_i(t-\Delta t)}{2\Delta t} \end{aligned}$$

in which the integration errors scale as $\mathcal{O}(\Delta t^4)$ for the positions. The velocity calculation is one time step behind the position calculation, and the velocities have errors of order $\mathcal{O}(\Delta t^2)$, but those errors do not accumulate because the velocity is not used to update the positions. The **velocity Verlet** [13] algorithm further improves this to $\mathcal{O}(\Delta t^3)$.

Molecular dynamics simulations of moderate size currently can access timescales on the order of a few microseconds. This means that processes that are inherently slower than this are poor candidates for study by molecular dynamics. Contrast this with Monte Carlo (next section), where the simulation is not constrained to follow a physically realistic trajectory.

The accuracy of the integration is dependent on both the finite accuracy with which quantities can be represented in the computer, and the errors introduced by the integration scheme itself. Since nearly all studied systems are chaotic, the simulation trajectory will diverge exponentially from the real trajectory and it has been argued that this compromises the usefulness of the technique. In practice this objection appears to have little bearing on the results for real systems. Although the trajectory is not a 'true' trajectory, it is a 'representative' trajectory.

The accumulation of numerical and integration errors can lead to drift in what should be conserved quantities such as momentum and energy and it is often necessary to thermostat long simulations. Thermostatting can also be used more broadly to simulate beyond the natural microcanonical ensemble (NVE) for molecular dynamics by allowing the energy the fluctuate to maintain a constant temperature (for example). Thermostatting inevitably further perturbs the trajectory from a physically realistic one, but again similar comments to the above apply. As long as the changes are small, the simulations provide useful answers.

Symplectic integrators [14] are a class of integration schemes where it can be proved that, despite the discretisation, the trajectories conserve exactly an underlying Hamiltonian. The Hamiltonian is not exactly the same as the one used to set up the simulation, but rather a small (unknown) perturbation of it. Advantageously, there are no integration errors at all. While numerical round-off errors remain; in practice, short to moderate length simulations can be successfully completed without resorting to thermostatting.

1.7 Monte Carlo

Monte Carlo (or the **Monte Carlo method**) contrasts with molecular dynamics in that it does not attempt to reproduce the real dynamics of a system, rather to approximately evaluate equation 3 by integrating it stochastically. This integration yields a non-physical trajectory through phase space. The details are discussed in subsequent sections, but generally suffers *less* from problems of ergodicity and rounding errors, and is unaffected by drift or integration errors. However it does not provide dynamical quantities, at least not directly. The freedom from having to choose a physical trajectory can aid sampling greatly where the physical process is slow compared to accessible simulation timescales. This is certainly true for the phase equilibria problems considered in this thesis.

Perhaps the most obvious way to numerically evaluate the integral of equation 3 is to

generalise the rectangle rule, and integrate over a hyperlattice spanning the phase space. This is fatally flawed. To see why we will examine the typical behaviour of the function $\mathcal{P}(\Gamma)$: the overwhelming majority of the phase space corresponds to microstates which make a negligible contribution to Z. We will call this majority of microstates which are of negligible statistical weight **non-contributing**. Any simple sampling scheme will necessarily spend much of its effort sampling these non-contributing microstates, and failing to sample those that do. Additionally the microstates on the sampling lattice will be redundant with other samples corresponding simply to a permutation of (identical) particles amongst lattice sites, leading to no new physics.

We can marginally improve our results by eliminating the redundancy problem and sampling at randomly selected points in phase space rather than a regular lattice:

$$\langle A \rangle \approx \frac{\sum_{i=1}^{M} A(\Gamma_i)\mathcal{P}(\Gamma_i)}{\sum_{i=1}^{M} \mathcal{P}(\Gamma_i)}, \tag{12}$$

where Γ_i are chosen uniformly over the phase space. This is known as **random sampling** (or sometimes simply Monte Carlo) and is effective at evaluating badly behaved integrals in low dimensions.

Fundamentally, both of these methods fail to sample the relevant parts of phase space. To improve the situation we appeal to **importance sampling** [15]. The idea is to bias selection of sampling points such that they are more likely to fall in contributing phase space. To this end, we define a weight function $w(\Gamma)$. If we can successfully generate samples with a probability distribution of $w(\Gamma)$ then we can approximately evaluate equation 3 as:

$$\langle A \rangle \approx \frac{\sum_{i=1}^{M} \mathcal{A}(\Gamma_i)\mathcal{P}(\Gamma_i)/w(\Gamma_i)}{\sum_{i=1}^{M} \mathcal{P}(\Gamma_i)/w(\Gamma_i)} \tag{13}$$

where the sample microstates $\Gamma_1, \Gamma_2, \ldots, \Gamma_M$ are drawn with probability density $w(\Gamma)$. If we consider the ideal case when $w(\Gamma) = \mathcal{P}(\Gamma)$ then it reduces to simply:

$$\langle A \rangle \approx \frac{1}{M} \sum_{i=1}^{M} \mathcal{A}(\Gamma_i). \tag{14}$$

But in general it is not possible to generate configurations sampled from a complex function such as $\mathcal{P}(\Gamma)$. (In fact any method which attempts to do so from first principles will require knowledge of Z as a prerequisite, creating a 'chicken and egg' problem.)

This problem was overcome, or at least circumvented, by Metropolis with his eponymous sampling scheme [15], and Hastings' generalisation to other distributions [16]. The algorithm exploits **locality** in $\mathcal{P}(\Gamma)$, i.e. the premise that if Γ contributes, then a nearby microstate in phase space, Υ, is also likely to contribute. To do this we must

surrender the independence of the samples. Each successive sample is now generated as a random perturbation of the previous one. The sequence of samples form a **Markov chain**, that is, the probability of generating a given microstate is dependent on the previous microstate but not (directly) on any other "history" of the system (such as the microstate two samples in the past). Markov chains are said to be **memoryless**.

Introducing this correlation greatly increases the chance that the new microstate will be contributing and thus improves the sampling. However since non-adjacent samples share a path of small steps between them, they are likely to be correlated themselves: only samples separated by sufficiently many steps so that the simulation has had time to wander far from the original microstate can be judged to be independent. A simulation is said to be **ergodic** if it is possible for a Markov chain to reach any microstate starting from any other. While many systems can be proved to be ergodic *in principle* (e.g. soft spheres under translations are trivially so), more relevant to real simulations is the problem of **practical ergodicity**, that is: is the simulation able to freely explore the phase space on accessible simulation timescales? Indeed, the time it takes for the system to fully de-correlate determines how accurate the answer is, because it effectively tells us how many truly independent samples we have collected. There is no full-proof method for demonstrating practical ergodicity, we have to rely on indirect evidence. There always remains the possibility that we are missing a region of important phase space, and we have to use problem-specific knowledge to assess the likelihood of this happening.

A Markov chain is any process which generates its next state according to a probability distribution that depends only on its most recent state. We can represent the probabilities as a 'matrix' π, where $\pi(\Gamma \to \Upsilon)$ is the probability of generating microstate Υ given that we are in Γ already. We can then impose the constraint we require, namely that, in the limit of many iterations of the Markov chain, the probability to be found in any state should be $\mathcal{P}(\Gamma)$. This also implies ergodicity since we impose this same limit irrespective of the starting state. Although we are relinquishing some of the flexibility, it nonetheless proves expedient to apply the condition of **detailed balance** to π, this says that:

$$\mathcal{P}(\Gamma)\mathrm{d}\Gamma\,\pi(\Gamma \to \Upsilon) = \mathcal{P}(\Upsilon)\mathrm{d}\Upsilon\,\pi(\Upsilon \to \Gamma). \tag{15}$$

It can be proved that this is a sufficient (but not necessary) condition for $\lim_{n\to\infty} \pi^n\Gamma_1 = \mathcal{P}(\Gamma)$ to hold, where Γ_1 is an arbitrary initial configuration [17].

Rather than attempt to write down values of π which satisfy this directly, it is usually easier to decompose the problem into *generating* and *accepting* a change:

$$\pi(\Gamma \to \Upsilon) = g(\Gamma \to \Upsilon)a(\Gamma \to \Upsilon), \tag{16}$$

where g is the probability of *proposing* a particular update and a is the probability of *accepting* that change. We must have $0 \leq a \leq 1$. This is a generalisation of the rejection method [18] for generating numbers drawn from an arbitrary probability distribution (used particularly where the distribution is not readily integrable). If we do not accept a particular update, then we must keep the old configuration, otherwise equation 15 would not hold. The decomposition has saved us from having to generate samples with a distribution like $\mathcal{P}(\Gamma)$. A typical update might be to move one particle in the system by choosing a random displacement selected uniformly over a cube centred on its current position. The update is then accepted with some probability satisfying equation 16. We can calculate the required probabilities using the detailed balance condition. Substituting $\pi = ga$ into equation 15 and re-arranging for the acceptances gives:

$$\frac{a(\Gamma \to \Upsilon)}{a(\Upsilon \to \Gamma)} = \frac{\mathcal{P}(\Upsilon)\, g(\Upsilon \to \Gamma)}{\mathcal{P}(\Gamma)\, g(\Gamma \to \Upsilon)}. \tag{17}$$

There are many ways of satisfying the above with keeping $0 \leq a \leq 1$, but by far the most common, and generally most efficient, is that originally chosen by Metropolis [15], namely:

$$\frac{a(\Gamma \to \Upsilon)}{a(\Upsilon \to \Gamma)} = \min\left\{1,\ \frac{\mathcal{P}(\Upsilon)\, g(\Upsilon \to \Gamma)}{\mathcal{P}(\Gamma)\, g(\Gamma \to \Upsilon)}\right\}. \tag{18}$$

The choice of updates is important to the efficiency of the simulation: large proposed changes lead to more independent samples but may lead to a poor acceptance rate. In practice we tune our moves (for example by adjusting the size of the cube we select over) to maintain a reasonable acceptance rate. A heuristic of 40% acceptance is commonly used [19].

We have now reviewed the machinery necessary in order to perform basic Monte Carlo simulations, which will be expanded and built upon in subsequent sections.

1.7.1 Pseudo-random number generators

Since Monte Carlo depends so heavily on random numbers, we must consider the quality (that is randomness) of the generators we use. There have been some famous failures [23, 20].

All the results presented here were obtained using the `ranvec` generator [21, 22]. This generator is known to have a defect in the nine point correlation function [24] but that was judged to be sufficiently obscure to be unimportant. In any case it has been well tested and serves as a good basis for comparison. Standard statistical tests [25] were used to confirm the implementation. Simulation results for a test case were also

compared against `ran2` [26] showing agreement within statistical bounds.

1.8 Barriers in computer simulation

Barriers are a universal concept in computer simulation, simply an impediment to a simulation visiting, or crossing, a particular value of some observable. More formally, we can define them as resisting movement through (or between) regions of phase space, and this is the usage adopted in this thesis. We can make a broad decomposition of barriers into three types. Barriers in real systems are often blends of the aspects below:

Energetic barriers are the simplest impediment to progress. We wish the simulation to pass through regions of high energy. The example of a classical particle undergoing Brownian motion in a double well potential demonstrates this type of barrier.

Entropic barriers are when we wish the simulation to preferentially access (or pass through) regions of small phase space volume (as a function of some order parameter of interest). For example, the fact that an Ising model[1] is unlikely to access the macrostate with all the spins aligned (since it corresponds to a single microstate) is because of entropic forces. We can generically refer to barriers that are part energetic and part entropic as **free energy barriers**.

Kinetic barriers are less fundamental, in that they are essentially an artifact of the particular (artificial) dynamics in use. In non-trivial systems, even when the energetic and entropic barriers are small (or have been defeated via extended sampling, see §2.1.3), the simulation will still not necessarily explore the full accessible phase space very quickly. This limit originates in the locality of our Monte Carlo moves: they each represent a small perturbation to the microstate. These perturbations will not necessarily add up to a 'large' change in the configuration necessary for full sampling. Kinetic barriers really become problematic when the accessible phase space is itself divided by small energetic or entropic barriers.

The term 'energy barrier' is sometimes used casually to refer to a 'free energy barrier'. This thesis will avoid such usage.

Clever methods of circumventing kinetic barriers are being developed all the time, a case in point is that of geometric cluster moves in fluids [27]. Here highly non-local

[1] a regular lattice of magnetic spins than are each wholly 'up' or 'down' and interact only with their four nearest neighbours

moves are used to quickly connect disparate regions of phase space and so yield highly efficient sampling by reducing the amount of 'kinetic' work needed to transition between the regions.

The extended sampling schemes introduced in chapter 3 allow us to completely circumvent pure energetic barriers. They also allow us to pass through entropic barriers, but only at the cost of reducing the acceptance rate of the Monte Carlo updates.

Chapter 2

Methods in phase equilibria

In this chapter, we review simulation techniques that have been developed to evaluate phase equilibria. This is still a very active research area, arguably because no universal, robust and efficient method has yet come to light. There has been considerably more progress in the case of liquid–vapour, liquid–liquid and glassy transitions than the fluid–solid or solid–solid cases; principally this stems from the long range order found in crystalline solids and the consequent symmetry-breaking nature of the transitions. This point is developed through examples in the following sections.

2.1 General methods

Before reviewing the methods themselves, it will prove expedient to present some underlying techniques which are now widely used in Monte Carlo studies, and in several cases form components of the individual coexistence methods. Extended sampling, in particular, is central to the studies later in this thesis.

2.1.1 λ-integration

λ-integration [28, 29, 30] was originally introduced as part of the thermodynamic integration method (see §2.2.1), but deserves independent attention. A generic parameter λ of the Hamiltonian is introduced. This may be a field such as volume or particle number, the temperature, or a parameter varying the form of the interaction itself. The method allows us to evaluate the free energy difference between two λ values, λ_1 and λ_2, by re-expressing the free energy derivative with respect to λ as an ensemble

average of a mechanical quantity.

In the below derivation $\mathcal{H} \equiv \mathcal{H}(\Gamma, \lambda)$, $Z \equiv Z(\lambda)$ and β is not necessarily constant (we may have $\lambda \equiv \beta$). Partially differentiating the free energy, $\beta F = -\ln Z$ with respect to λ, then substituting the definition of Z from equation 2 gives:

$$\begin{aligned}
\frac{\partial(\beta F)}{\partial \lambda} &= -\frac{1}{Z}\frac{\partial Z}{\partial \lambda}. \\
&= -\frac{1}{Z}\frac{\partial}{\partial \lambda}\int e^{\beta\mathcal{H}}\,\mathrm{d}\Gamma \\
&= -\frac{1}{Z}\int \frac{\partial(\beta\mathcal{H})}{\partial \lambda} e^{\beta\mathcal{H}}\,\mathrm{d}\Gamma \\
&= -\left\langle \frac{\partial(\beta\mathcal{H})}{\partial \lambda} \right\rangle_{\lambda}. \qquad (1)
\end{aligned}$$

For $\lambda \not\equiv \beta$ and $\lambda \equiv \beta$ respectively, we have:

$$\frac{\partial F}{\partial \lambda} = -\left\langle \frac{\partial \mathcal{H}}{\partial \lambda} \right\rangle \quad \text{and} \quad \frac{\partial(\beta F)}{\partial \beta} = -\langle \mathcal{H} \rangle . \qquad (2)$$

The free energy difference can then be estimated simply by integrating:

$$\Delta F \cong \int_{\lambda_1}^{\lambda_2} \frac{\partial F}{\partial \lambda}\,\mathrm{d}\lambda, \qquad (3)$$

where the free energy derivative is evaluated at regular intervals using Monte Carlo simulations to estimate the appropriate ensemble averages. To summarise: we have shown one method of calculating the free energy difference between (and thus the relative stability) of simulations at different points on the phase diagram.

2.1.2 Histogram reweighting

Histogram reweighting [31, 32] allows us to extract additional information from a Monte Carlo simulation. Results obtained with one sampling distribution, $\mathcal{P}(\Gamma)$ can be **reweighted** to give an estimate of results from a distinct, but similar, distribution $\mathcal{P}'(\Gamma)$. If we record details of the microstates visited during a simulation then we can first **unfold** (divide out) the sampling distribution used during the simulation, then apply our alternative one to the result. This is perhaps best illustrated by comparison of random sampling, straight importance sampling and histogram reweighting itself:

Method	Sampling distribution	Weight	Effective probability
Random sampling	1	$\mathcal{P}(\Gamma)$	$\mathcal{P}(\Gamma)$
Importance sampling	$\mathcal{P}(\Gamma)$	1	$\mathcal{P}(\Gamma)$
Histogram reweighting	$\mathcal{P}(\Gamma)$	$\mathcal{P}'(\Gamma)/\mathcal{P}(\Gamma)$	$\mathcal{P}'(\Gamma)$

In each case the sampling distribution is input to the simulation to control which microstates are sampled, and the weight is applied to each sample during the calculation of ensemble averages–usually as a post-process to the main simulation. The effective probability is the product of the sampling distribution and weight. To use the technique we thus replace equation 14 with a weighted average:

$$\langle \mathcal{A} \rangle' \cong \frac{\sum_{i=1}^{M} w_i \mathcal{A}(\Gamma_i)}{\sum_{i=1}^{M} w_i}, \tag{4}$$

where the weights are given by $w_i = \mathcal{P}'(\Gamma_i)/\mathcal{P}(\Gamma_i)$.

Histogram reweighting is commonly used to explore the phase diagram in the vicinity of a point at which we already have simulation results. For example: if we perform a simulation in the NPT ensemble at a prescribed pressure P and inverse temperature β, then the sampling distribution would be:

$$\mathcal{P}(\Gamma) = e^{-\beta(\mathcal{U}(\Gamma) - PV(\Gamma))}, \tag{5}$$

where $\mathcal{U}$ is the internal energy of the microstate and V the instantaneous volume. As long as the volume and energy of each of the sampled microstates is recorded, then we can estimate the results for any given observable by substituting equation 5 into $w_i = \mathcal{P}'(\Gamma_i)/\mathcal{P}(\Gamma_i)$, yielding:

$$w_i = e^{(\beta' P' - \beta P)V(\Gamma_i) + (\beta - \beta')\mathcal{U}(\Gamma_i)}. \tag{6}$$

This formula is used to analyse results obtained in chapter 6.

The validity of the extrapolation depends on the similarity of $\mathcal{P}$ and $\mathcal{P}'$. As we move away from the phase space which was adequately explored originally, we find some samples are given increasingly disproportionate weights and so reduce the overall statistical quality of the results. The severity of this problem can be estimated by looking at the overlap of sample distributions in the $\mathcal{U}$-V plane (for the NPT ensemble), or by analysing the weight factors themselves. A useful indicator is the ratio:

$$r = \frac{\sum_{i=1}^{M} w_i}{M w_{\max}}, \tag{7}$$

where $w_{\max}$ is the maximum weight. The ratio r is unity when $\mathcal{P} = \mathcal{P}'$ and decays towards zero as the statistical quality falls. Approximately, it corresponds to the fraction of original samples which are still statistically relevant under the new distribution. A heuristic cut-off of around 0.2 proved useful in this work. A more formal error analysis can be found in reference [33].

Multiple histogram reweighting [31, 32] is an extension to histogram reweighting which allows data from multiple simulations at different points of the phase diagram to be integrated, further extending the range over which extrapolations can be made. Essentially, it operates by calculating the probability density each sample would have been assigned to each of the other simulation statepoints, and thus the ratios of the partition functions between the different simulations can be estimated. Multiple histogram reweighting can be computationally expensive, particularly when finding these partition function ratios.

2.1.3 Extended sampling

Extended sampling [34, 35], also known as **multicanonical sampling**, or the method of **expanded ensembles** is a key technique in Monte Carlo simulation. It generalises the earlier **umbrella sampling** [36, 37]. The method uses a biased sampling distribution, introducing an additional contribution distinct from the usual Boltzmann factor:

$$\overline{\mathcal{P}}(\Gamma) = e^{-\eta(\Gamma)} e^{-\beta \mathcal{H}(\Gamma)}, \tag{8}$$

where η is known as the **multicanonical weight function**, and is typically defined as a function of one or more macrovariables, which we will refer to as an order parameter. Analogously to histogram reweighting, the weight function bias can be **unfolded** from the results:

$$\langle \mathcal{A} \rangle = \frac{\sum_{i=1}^{M} e^{\eta(\Gamma_i)} \mathcal{A}(\Gamma_i)}{\sum_{i=1}^{M} e^{\eta(\Gamma_i)}}. \tag{9}$$

The difference from histogram reweighting is really a matter of intent rather than methodology: we use extended sampling to set up a simulation with a 'biased' sampling distribution where the bias will later be removed, whereas histogram reweighting is used to extrapolate from existing data. Both techniques can be combined in a single simulation (see later chapters).

There are multiple applications for extended sampling:

- the energy barrier in a double well potential can be overcome by enhancing the probability of visiting the barrier states;

- in some cases we can generalise this to free energy barriers, where a suitable order parameter which can drive us through the transition exists;
- even more generally, we may be able to construct an artificial route through phase space (and a corresponding order parameter) and again penetrate the free energy barrier;
- finally, it may allow reference states (such as very low density) to be reached, which can be useful for linking numeric Monte Carlo data to analytical results which only apply in certain limits.

Both λ-integration and extended sampling can be used to connect states, the difference is that extended sampling does so within a single Monte Carlo simulation. Extended sampling is discussed in detail in chapter 3, and is central to the later phase switch Monte Carlo studies of chapters 5 and 6.

2.2 Phase coexistence methods

All the methods presented below can be used to locate phase boundaries in a variety of systems, and under different circumstances. Extended sampling is applied directly to locate liquid-vapour coexistence for the Lennard-Jones system in chapter 4. The **phase switch** or **lattice switch** method is reviewed, and applied, in later chapters.

2.2.1 Thermodynamic Integration

Thermodynamic integration is one of the oldest coexistence methods [29], but is far from a museum piece. Its wide applicability and simplicity have meant it is still a staple method for determining coexistence properties. The basic idea is very simple: λ-integration (described in §2.1.1) is used to link each phase to a reference state for which the free energy can be calculated analytically. We choose λ to be an appropriate control parameter for this purpose. Typical reference states are the single occupancy cell [29] and Einstein crystal [39] (for crystalline solids), and the ideal gas (for liquid and vapour states). In practice, it is not necessary to reach the reference state itself, corrections can be applied to states that are 'sufficiently ideal' [39, 40]. The free energy differences between the reference and experimental states are determined entirely independently for each phase. The free energy difference then follows from comparison of the two 'branches' of the calculation.

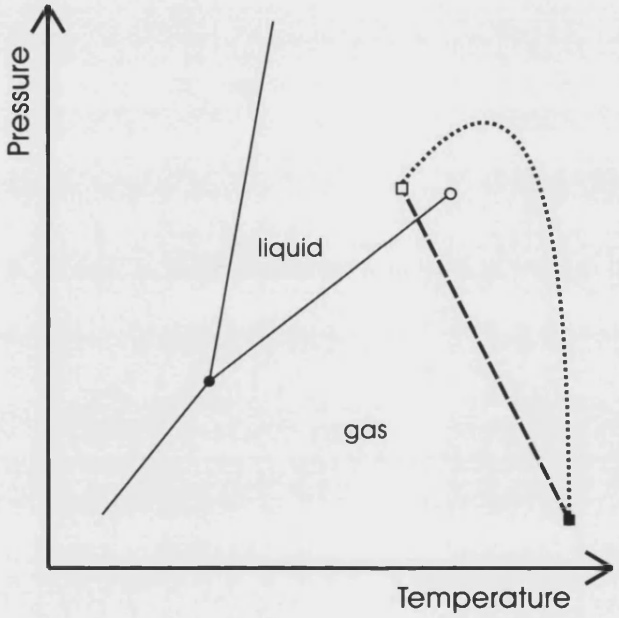


Figure 2.1: Two possible integration paths between an ideal gas like reference state and a liquid phase state for a simple fluid. The dotted path crosses the liquid–vapour transition hence would not yield a valid result.

Thermodynamic integration relies on expressing the control parameters of the ensemble as free energy derivatives. For the NPT ensemble, we use:

$$\left(\frac{\partial F}{\partial P}\right)_{NT} = V \qquad \text{and} \qquad \left(\frac{\partial(\beta F)}{\partial \beta}\right)_{NV} = \mathcal{H}. \tag{10}$$

Control parameters can always be expressed in terms of appropriate free energy derivatives, so the method is applicable to any ensemble.

Singularities occur in the free energy derivatives at first order phase transitions, so crossing such a boundary will give incorrect results. A simple example is shown in figure 2.1. The need to reach reference states which are usually a long way from the boundary of interest makes thermodynamic integration a laborious method. The long path leads to accumulation of integration errors. Further, there is no general technique to derive an accurate error bound.

2.2.2 Gibbs-Duhem integration

Gibbs-Duhem integration [41, 42, 43] is an attractive alternative to thermodynamic integration if we already have one point on the coexistence curve. We proceed from the Clausius-Clapeyron relation. For a first order phase transition between two phases (denoted A and B) it can be expressed as:

$$\frac{\mathrm{d}P}{\mathrm{d}T} = \frac{S_A - S_B}{V_A - V_B}. \tag{11}$$

At the phase boundary, the free energy of the two phases will be equal, so substituting from $F = U - TS$, we find that:

$$\frac{\mathrm{d}P}{\mathrm{d}T} = -\frac{1}{T}\frac{U_A - U_B}{V_A - V_B}. \tag{12}$$

If we know one point (P_0, T_0) on the phase boundary, then we can estimate U_A and V_A from a single phase simulation in phase A, and U_B and V_B from an entirely separate simulation in phase B, allowing evaluation of the gradient of the phase boundary in terms of single phase averages only. It is then simply a case of integrating equation 12 to yield the entire boundary. Larger steps along the curve can be made if the technique is combined with histogram reweighting (see section 2.1.2). Although it was derived here for the NPT ensemble, similar results hold in other ensembles, proceeding from the generalised Clapeyron equation.

The chief criticism of the method is a lack of feedback. There is no way to confirm that the integration does not accumulate errors, as there is a wide band of metastable

states around the phase boundary. The validity of the method is also contingent on the accuracy of the original state point, which must be obtained independently. The ability to work entirely in terms of single phase averages leads to commendable computational efficiency however, and Gibbs-Duhem integration has been successfully applied in a number of studies [44, 45, 46].

2.2.3 Gibbs Ensemble methods

The **Gibbs ensemble method** of Pangiotopoulos [47, 48] simulates both phases in separate simulation boxes, but unlike in Gibbs-Duhem integration, connects those boxes thermodynamically. Particle transfers and volume exchanges between the boxes are introduced, providing for equilibration. Total volume and particle number are conserved, yielding an NVT ensemble comprising the two sub-systems. Such an approach works well for equilibria amongst low density fluids but the particle transfer moves suffer low acceptance in dense fluids. The method becomes impractical for solid phases.

The **virtual Gibbs-Ensemble** method of Escobedo [49] is an attempted generalisation which relaxes the above constraints. The particle transfers and volume exchanges are adapted to modify only one of the boxes (e.g. a particle transfer becomes a particle deletion, but with the energy change it would have caused in the target box entering into the acceptance). The 'faked' moves violate detailed balance. However, with suitable assignment of phases and types of moves the simulation can be constructed that stably converges on coexistence, with the violation actually driving the simulation in the correct direction [50]. It should be noted that no theoretical framework has so far validated the method in light of the violation of detailed balance, but empirical evidence suggests that the 'steady state' results it produces are equivalent to those of traditional Monte Carlo simulations.

A further modification is still required if crystalline solid phases are to be studied. Since even virtual swaps cannot occur to or from a solid it is necessary to simulate an explicit interface between the solid and a fluid. One box now contains a pure fluid phase whereas the other box is elongated and initialised with a 'slab' of crystalline solid embedded in a fluid (still under periodic boundary conditions). Virtual moves serve to equilibrate the fluid contained in this box with the pure phase. This method was applied to the Lennard-Jones system by Shetty [50], however very low acceptance rates were recorded for the virtual particle exchanges making it unclear if there was any advantage over discarding the pure liquid phase and conducting a simple explicit simulation of coexistence. In any case, the simulation will suffer from interfacial effects in the crystalline phase necessitating the use of a large simulation box.

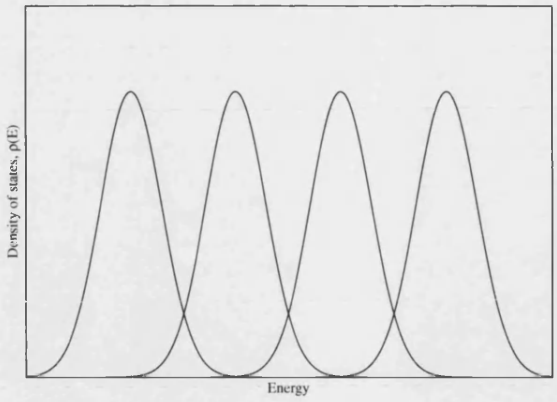


Figure 2.2: Series of energy distributions, one for each of the temperatures covered by a parallel tempering run. In order for configuration swaps to be accepted between adjacent boxes, it is necessary for the respective energy distributions to overlap.

2.2.4 Parallel tempering

Parallel tempering [51, 52, 53, 54, 55], also known as **parameter hopping** and **replica exchange**, has generated considerable interest where acceleration of processes with slow kinetics is desired. The idea is to simulate at a range of different temperatures, from the subject temperature up to a higher one at which the dynamics are substantially faster. All the simulations are run in parallel, and swaps of the entire configuration are attempted between adjacent temperatures on the chain (or, equivalently, we can exchange temperatures and labels). These swaps are performed with the usual Metropolis probability based on the net change in energy after the trial swap is performed:

$$\mathcal{P}(\Gamma \to \Upsilon) = \frac{e^{-\beta_2 \mathcal{H}_1} e^{-\beta_1 \mathcal{H}_2}}{e^{-\beta_1 \mathcal{H}_1} e^{-\beta_2 \mathcal{H}_2}}, \tag{13}$$

where Γ has configuration 1 of energy $\mathcal{H}_1$ in the box of temperature β_1 and similarly for configuration 2, whereas Υ reverses this.

Generally the box at higher temperature will decrease energy slightly (as it receives a 'cooler' configuration) and the cooler box will increase energy substantially (as it receives a 'hotter' configuration, and the lower temperature alters the Boltzmann distribution to penalise inter-penetrations more harshly). Roughly speaking, the acceptance rate is determined by the degree of overlap between the two boxes' energy histograms (see figure 2.2). These swaps allow configurations to (eventually) migrate all the way from the bottom to the top of the chain, since the simulation can equilibrate to the local temperature at each step. Thus subject temperature configurations will get a chance to undergo rapid dynamics at higher temperatures before returning to their original

one. This rapid evolution can be used to speed up the sampling of phase transitions.

A further advantage is obtained if the higher temperature data is also of use. Indeed, given that the simulations must support a continuous energy spectrum, we can use histogram reweighting (see section 2.1.2) to obtain estimates of observables at any intermediate temperature. The technique is also an excellent candidate for a parallel processing implementation in that inter-box swaps can be made infrequently, minimising synchronisation requirements and so maximally exploiting the parallelism.

If in a particular system the probability of accepting a swap between non-adjacent temperatures is significant, then it may be advantageous to attempt these swaps as well. Parallel tempering can be generalised to depend on some other parameter than temperature: the pressure or even the potential itself could be varied, provided the changes were gradual enough to retain overlapping energy histograms.

2.2.5 Constrained fluid λ-integration

Constrained fluid λ-integration is another technique based on λ-integration, proposed by Grochla [56] who demonstrated it on the Lennard-Jones system. It was later combined with Gibbs-Duhem integration (see §2.2.2) by Elke *et al* [57] and applied to NaCl crystals. Rather than varying the control parameters, the method seeks to take the material directly through the transition by manipulating the system potential itself. This is accomplished in a number of stages, mapped to increasing λ values. Here we consider a single crystallisation transition:

1. The interaction potential is gradually weakened, speeding up diffusion within the fluid.
2. Next, a series of Gaussian potential wells are gradually introduced at fixed points in space lying on a crystalline lattice of the target structure. Simultaneously, the volume is reduced from the typical fluid value to that of the solid. Under the conditions of increased density, weak interactions and Gaussian wells it is hoped that the particles will rearrange into the chosen crystal structure, and do so much faster than a real system would undergo the transition.
3. In the final stage the full potential is restored at the same time as the Gaussian wells are switched off.

In a real simulation, the system will not proceed simply up and down the list but rather perform a random walk in the λ parameter.

Conceptually this method offers an immediate advantage over straight thermodynamic integration: the inter-phase path does not depend upon distant idealised reference states. It attacks the phase boundary directly–there is a single path linking the two states of interest from which the free energy difference follows directly.

The most challenging part of the simulation is solving the "diffusion problem": getting the particles to distribute evenly amongst the lattice sites. Fundamentally, there is no driving force to match wayward particles to unoccupied lattice sites. It seems likely that the poor scaling behaviour of this aspect will in practice preclude application to larger systems. A further criticism is the length and complexity of the inter-phase route; the significant weakening of the potential is analogous to a large increase in temperature, rendering the route more indirect than it initially appears. A path of this complexity might also be at risk from artificial singularities. This apparent lack of robustness or scalability leads us to look for other methods.

2.2.6 Mastny and de Pablo method

Mastny and de Pablo [59] have proposed a method which aims to directly estimate the free energy difference of the two phases. The rationale for the method is the authors' assertion that: "To connect the free energy of the solid and liquid phases, there must exist a portion of energy and volume space that can be simultaneously sampled by both solid and liquid phases". To exploit this premise, a series of simulations of the Lennard-Jones system were performed, each restricted to, and sampling the configuration space of, one of a set of overlapping "windows" in the energy–volume (E-V) plane. Successive windows were positioned to form a path linking the two well-separated regions of E, V associated with the typical configurations of the respective phases. Along this path, a central range of E, V was indeed reported to be found for which both liquid and solid phase configurations could be sampled. A simple average of the solid phase and liquid phase density of states was accumulated in the central range and joined for continuity to the measured forms of the liquid and solid density of states on either side.

The existence of a range of E and V that can be sampled by both liquid and solid phases appears insufficient to connect the respective branches of the density of states because these are (like their underlying sets of characteristic configurations), fundamentally *distinct*. Instead, for a proper connection, the phases must be linked via a continuous (and reversible) path through configuration space. In the context of the method of Mastny and de Pablo, this necessitates repeated (back and forth) *transitions* between the two pure phases. Since no such transitions were observed [60], the validity of the method appears questionable.

2.2.7 The multi-NPH method

Another approach to the fluid–crystalline solid coexistence problem, which bears some resemblance to that of ref. [59], is the "multi-NpH" method of Escobedo [61]. Here a path is constructed in the *enthalpy* of the system, spanning the range of values between those typical of the respective equilibrium phases at some prescribed pressure. The system temperature is ascertained at each point along the path, and exhibits a discontinuity at some value of the enthalpy as the favoured phase of the system changes. Then λ-integration with respect to the temperature variations along the path yields the free energy difference. One problem reported with this method was that it was apparently necessary for equilibration purposes to initialise the system in the solid phase at all points along the enthalpy path. However, it was difficult to be sure whether, for a given enthalpy value, the system had relaxed to the phase of minimum free energy, and hence whether the integration result was unbiased. This appears to be a manifestation of the familiar drawback of thermodynamic integration, namely the difficulty of finding a reversible integration path.

Chapter 3

Extended sampling

Extended sampling is discussed in detail in this chapter as it underlies much of the remainder of the thesis. The most challenging part of applying extended sampling is determining a weight function, so we devote space to the variety of methods that have been developed for this purpose. The choice of algorithm will prove crucial for achieving useful results in an acceptable time-span.

3.1 Introduction

Extended sampling was introduced in §2.1.3, but is discussed in detail in this chapter, as it is fundamental to the studies later in this thesis. It is usually applied in one of these contexts:

- overcoming free energy barriers separating interesting regions of phase space;
- accessing regions of phase space which are of negligible statistical weight, but are interesting in the context of the system studied;
- *or*, allowing a single simulation to sample regions of phase space associated with a range of values of the applied fields rather than single values.

Simple energy barrier type problems, such as the double well potential shown in figure 3.1 (a) can be solved almost perfectly–a weight function can be derived which alters the sampling to be approximately flat with comparative ease. The system will rarely cross from one basin to the other in either a molecular dynamics or importance sampled

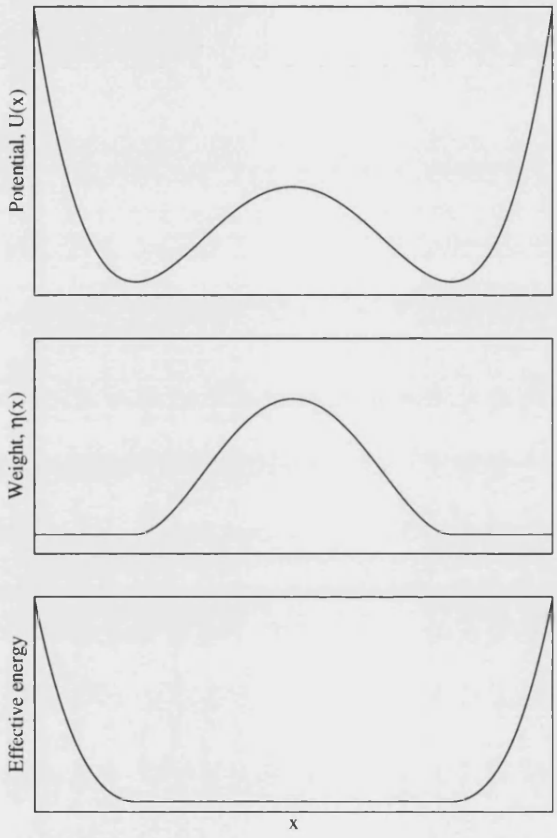


Figure 3.1: (a) The double well potential is shown. (b) A weight function designed to overcome the barrier. (c) The effective (simulation) energy distribution, after the weight function is applied.

Monte Carlo simulation. If better sampling is desired, we can overcome the barrier with a multicanonical weight function. By setting

$$\eta(x) = \ln \mathcal{P}(x) = \ln e^{-\beta U(x)} = -\beta U(x),$$

and only doing so over the interval of x between the wells (applying a flat weight function elsewhere) we produce the graph of figure 3.1 (b). This weight function is then used to bias the simulation such that it no longer encounters the barrier. We simulate with the modified sampling distribution:

$$\mathcal{P}(\Gamma) = \frac{1}{\tilde{Z}} e^{-\beta U(x) - \eta(x)}. \tag{1}$$

The simulation effectively sees the potential shown in figure 3.1 (c) and will now be free to cross the barrier. The bias can be removed by reweighting the results, as per equation 9. In higher dimensions it is unlikely that an order parameter (x in this case) will be either so readily identified or effective at penetrating the barrier.

Extended sampling also allows us to attack problems of the phase space volume (entropic) kind but will suffer from a slowdown as the acceptance rate drops in steep parts of the weight function; nor is it likely to assist in overcoming kinetic barriers. Extended sampling does interact with the artificial dynamics: if we apply preweighting which lowers a barrier separating two free energy wells in the system, we may speed up the sampling of the entire relevant phase space considerably. On the other hand, extended sampling may exacerbate an existing kinetic barrier, or the expansion of accessible phase space may introduce new barriers, and with them the potential for new ergodicity problems.

It is possible to combine an extended sampling scheme with molecular dynamics simulation; **Metadynamics** [62], for example, does exactly this. Such a simulation combines semi-physical trajectories with the barrier penetration potential of Monte Carlo, though it does so at the cost of having neither the sound theoretical framework of statistical mechanics or realistic dynamics.

3.1.1 The weight function

Given a system with a Hamiltonian, $\mathcal{H}(\Gamma)$, we derive a modified sampling distribution, $\widetilde{\mathcal{P}}$, that samples an expanded ensemble containing the same physics as $\mathcal{H}$, as well as accessing additional regions of phase space, or (rarely) suppressing existing ones. We do this by introducing the multicanonical weight function $\eta(\Gamma)$. The modified probability

distribution (see §2.1.3) is then:

$$\widetilde{\mathcal{P}}(\Gamma) = \frac{1}{\tilde{Z}} e^{-\beta\mathcal{H}(\Gamma)-\eta(\Gamma)}. \tag{2}$$

It follows directly that the statistical weight of a microstate sampled from a simulation under equation 2 can be related to the equilibrium equivalent (see equation 1 on page 16) using:

$$\mathcal{P}(\Gamma) = \frac{\tilde{Z}}{Z} e^{\eta(\Gamma)}\, \widetilde{\mathcal{P}}(\Gamma). \tag{3}$$

Thus we can recover the physics given by the original $\mathcal{H}$ by applying equation 3 to the samples before computing averages. This procedure is generally known as **unfolding the weights**.

3.1.2 Selecting an order parameter

To make this useful we must assign a weight to each microstate of the system. Typically this is done via the introduction of one or more order parameters. The choice of order parameter is key. Without loss of generality, consider linking two regions of phase space which the simulation would not normally sample. Important considerations are:

- the length of the route–shorter is better;
- the ability of the specific Monte Carlo moves to move in order parameter space (kinetics);
- that it identifies phase space of interest easily;
- the steepness of the weight function–generally we want to bound $\mathrm{d}\eta/\mathrm{d}M$ (where M is the order parameter) so our binning strategy is not undermined (see below).

In some circumstances it can be useful to use multiple orthogonal weight functions ($e^{-\eta(\mathcal{A})}e^{-\nu(\mathcal{B})}$) or a fully two dimensional sampling ($e^{-\eta(\mathcal{A},\mathcal{B})}$), examples are given in references [59] and [63].

We will denote a generic order parameter as M. Although not necessitated, it is a common practice to discretise (or 'bin') the order parameter and use a piecewise weight function which is easy to store and reference. We term the ensemble of microstates belonging to a given order parameter bin a macrostate; and define the statistical weight of a macrostate as a partition function constrained to only include those microstates belonging to the macrostate. Usually it is sufficient (or in some cases necessitated)

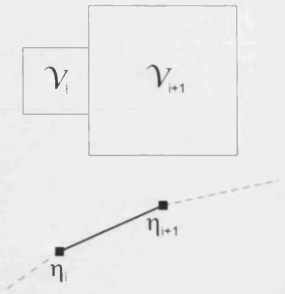


Figure 3.2: Schematic of two adjacent macrostates of different phase volumes against an order parameter.

to assign a single uniform weight to the entire macrostate, but various interpolation schemes could also be used. Alternately, the weight function could also be expressed analytically if appropriate.

3.1.3 Entropic slowing down

Consider a one dimensional, discretised order parameter, M_i. The phase space volume, v_i, of each macrostate increases with the index, so $v_{i+1} > v_i$. Now if we wish to bias the simulation so that it visits two adjacent macrostates equally via a weight function, η_i, we can simply set $\eta_i = \ln v_i$ (cf. equation 2). In general we will not know the phase space volumes *a priori* so the weight function must be found iteratively (this is discussed in the following section). The arrangement it shown schematically in figure 3.2. The acceptance rule for a move from macrostate i to $i+1$ would be:

$$a(i \to i+1) = \min\left\{1, e^{\eta_i - \eta_{i+1}} e^{\beta(\mathcal{H}_i - \mathcal{H}_{i+1})}\right\}. \tag{4}$$

The expression $e^{\eta_i - \eta_{i+1}}$ becomes v_i/v_{i+1}. Since the energy factor will in general fluctuate largely independently of the weight function factor we can see that this represents a reduced acceptance, in proportion to the ratio of the phase space volumes. Conversely: consider a move from macrostate $i+1$ back to macrostate i. In this case the acceptance is likely to be high because of the weight function factor, but the probability of generating such a move will again be v_i/v_{i+1}. In both cases the transition probability, $\pi = ga$, will be suppressed by the same factor and our simulation will necessarily suffer low acceptance. The argument is more general than it appears: the rule is transitive and the macrostates do not have to be adjacent, or ordered as implied above. This low acceptance is fundamental: it would occur even in an ideal gas system where the Hamiltonian does not enter into the acceptance (at least in the NPT ensemble). One might think that you could artificially target macrostates of low phase volume with

a move with a biased generation probability targeted to regions of lower phase space volume. However, such a bias necessarily introduces a compensating factor into the acceptance probability which cancels out the bias, so that the integrity of the underlying Markov chain is preserved (fundamentally this stems from the incompressibility of phase space). We call this effect **entropic slowing down**, it represents a lower bound on the number of trial moves necessary to penetrate a given barrier (sometimes known as the "well depth").

3.1.4 Determining weight functions

It is useful to have a simple criterion to aim for when finding weight functions. Clearly, if the simulation visits the entire range of interest with reasonable statistics in all regions, then we will have the data we need to properly estimate observables. Therefore a criterion of **flatness** suggests itself [64]: if we visit all values of the order parameter with equal frequency then the data we collect *will* eventually give us the statistics we need. Therefore flatness implies efficacy, but does not necessarily imply efficiency. Nevertheless, flat histograms are a common aim in existing studies.

It is useful to introduce the concept of **local diffusivity**, this is the rate at which the simulation de-correlates as a function of the order parameter. We refer to the simulation as "fast" if it de-correlates quickly and "slow" if it does not. There are many ways of defining a local diffusivity. Troyer *et al* [65] suggested an effective one; unfortunately it requires a simulation that visits the entire order parameter range many times over just to come up with an initial estimate of the local diffusivity and so its practical utility is limited to cases where we are seeking a large quantity of data from one system (weight function). It goes on to make the reasonable assertion that we need to visit each point in order parameter space with probability inversely proportional to the local diffusivity, and derives and demonstrates a procedure for doing so. In many real problems we do not care if the weight function is optimal, merely that it is good enough. Additional time spent finding the weight function is less time generating the actual results of interest.

3.2 Algorithms

As stated, we can only hope to obtain a weight function with a refinement process: we need a method which continues to improve our estimate until we achieve the flat histogram or other convergence criteria. Happily there are numerous schemes to choose

from, each with a particular blend of benefits and shortcomings. As such, we proceed by identifying the characteristics which would be desirable in a weight function algorithm:

- **Outreach**: ability of an algorithm to sample in regions about which it has little knowledge; those on the limits of current exploration.
- **Equilibrium** dynamics: most algorithms operate at least slightly out-of-equilibrium, but the degree to which this is true is critical. Any results obtained under sufficient violation of detailed balance will be unreliable.
- **Utilisation**: some algorithms only use samples collected from a subset of the total simulation effort so far. Others can benefit from all of it.
- **Rejected moves**: algorithms such as 'waste recycling' and the 'transition matrix' method can derive useful data from rejected moves as well as accepted ones.
- **Robustness**: algorithms that require detailed tuning to ensure convergence are best avoided.
- **Simplicity and transparency**: obviously these qualities are desirable.

3.2.1 Visited states

The so called **visited states** algorithm [66, 67] is possibly the simplest method for determining weight functions, nevertheless it is powerful. It proceeds by collecting a histogram of frequency, $H(M)$, with which a normal simulation visits each macrostate over a reasonably long run. A system will typically only visit (sample) a restricted order parameter range (otherwise we would not have appealed to extended sampling in the first place). This data is then used as to set up a weight function $\eta(M) = \ln H(M)$ which corresponds to a flat histogram (cf. equation 2). Examination of acceptance rules such as equation 4, which are always ratios of sampling distributions, should convince one that the weight function is only significant within an additive constant.

Unfortunately, macrostates that were not visited frequently in the initial simulation will have poor (or non-existent) statistics, so the estimate of the requisite weight function will be of poor quality in outlying regions. But by making the scheme iterative, a little more of these outlying regions can be explored with each iteration, and eventually good quality statistics for the full interval obtained.

When constructing weight functions for later iterations the simulation is preweighted with the most recent estimate of $\eta(M)$, and hence measures a 'preweighted' frequency

histogram, $\tilde{H}(M)$ which must be unfolded to give the unbiased histogram, $H(M)$, before it can be used as an (improved) estimate of the weight function. The unfolded histogram for an iteration will have better statistics in the outlying regions than its predecessor.

The unfolding follows from equation 3, so $H(M) = e^{\eta(M)}\tilde{H}(M)$ (to within a constant factor). The iterative scheme becomes:

$$\eta_{i+1}(M) = \ln H_i(M) = \eta_i(M) + \ln \tilde{H}_i(M), \tag{5}$$

where the subscripts label iterations. The process is initiated with a flat weight function: $\eta_1(M) = 0 \quad \forall M$.

When the simulation does wander into regions of M for which the weight function has poor statistics, it will tend to get stuck on statistical barriers (noise) and make little outward progress. Thus it is advantageous to use a flat weight function in these outlying regions. To implement this practically, macrostates that were not visited more than a threshold number of times are marked as such and then flattened by cloning the value of the weight function from the nearest unmarked macrostate.

It is normal procedure to collect independent samples from a Monte Carlo simulation by sampling at regular intervals. The interval is often chosen to be quite large so the results can be judged to be statistically independent (in some sense). For the purposes of visited states, however, throwing away most of the information about which states were actually visited is wasteful. Accepting that the results are not statistically independent, sampling can be performed every Monte Carlo move. This is called **oversampling** and allows a much smoother histogram to be obtained.

A termination criterion should be added: once the full order parameter range has been explored, the weight function has been found and the algorithm can terminate.

The visited states algorithm's main advantages are simplicity and the fact it runs in an equilibrium simulation. When combined with windowing (see §3.2.4) it can be a formidable method. Unfortunately we need to determine a good length for each iteration by trial and error, but such a determination may only need to made once. The need to fully explore the accessible order parameter space before any of the current statistics can be integrated leaves an algorithm with room for efficiency improvements.

3.2.2 Wang-Landau

The **Wang-Landau** algorithm [64, 68, 69] takes a different approach than visited states: it operates significantly out of equilibrium, updating the weight function continuously but providing a means to eventually relax to equilibrium behaviour. Wang-Landau seeks to actively drive the simulation away from areas it has already sampled. The update schemes operates thus:

$$\eta(M) \leftarrow \eta(M) + f, \tag{6}$$

where f is the **modification factor**, often initialised to unity. This update is performed after every Monte Carlo move leading to a very rapid evolution of the weight function. Note that the update rule corresponds to a multiplication of the probability of the macrostate so large barriers can be overcome relatively quickly. Eventually the simulation will reach a steady state where the shape of the weight function is fixed and the simulation is visiting the whole of the order parameter space with equal probability. The residual shape of the $\eta(M)$ represents an estimate of the true equilibrium weight function.

Unfortunately the violation of detailed balance that Wang-Landau entails leads to feedback into the weight function itself. In practice the weight function will end up exaggerated compared to the equilibrium version.

Once steady state has been reached, we detect it by looking for a flat histogram of visited states. We use a definition that *each* macrostate should have a prescribed percentage as many visits as the mean number of visits across *all* macrostates. Once that condition is reached we perform an iteration: the existing weight function is kept but the modification factor, f, is updated. A number of iteration rules are possible, but the most common one is:

$$f \leftarrow \frac{f}{2} \tag{7}$$

After iteration, the dynamics will be closer to an equilibrium simulation because the detailed balance violation has been reduced. The steady state weight function will change as well, but the existing one will be a reasonable estimate. As f continues to decrease we eventually recover equilibrium dynamics, along with the final weight function. Convergence can be confirmed by disabling modification of the weight function and observing the histogram of visited states.

Wang-Landau has a unique ability to push into unexplored areas of the order parameter space which is an attractive feature, however the non-equilibrium dynamics involved cause problems of their own. The implications for the speed of convergence,

in particular, will be examined through case studies in later chapters.

3.2.3 Transition matrix

The **transition matrix** method [70, 71, 72] distinguishes itself from the other algorithms in two regards: it derives information from the *transitions* rather than just which states are visited and it also incorporates information gleaned from rejected moves. The idea rests on recording a matrix of the transitions between macrostates, distinct from the underlying microstate transition probabilities. Because the specific microstate the simulation is in effectively encodes extra information on top of the macrostate label, the system will not be a Markov chain because it is not memoryless. We proceed defining a collection matrix C which is updated after every Monte Carlo move:

$$\begin{aligned} \mathrm{C}(M_i \to M_j) &\leftarrow \mathrm{C}(M_i \to M_j) + a \\ \mathrm{C}(M_i \to M_i) &\leftarrow \mathrm{C}(M_i \to M_i) + (1 - a), \end{aligned} \tag{8}$$

where a is the acceptance probability. Such a scheme provides can provide an estimate of the underlying macrostate transition probabilities, Π:

$$\Pi(M_i \to M_j) \cong \frac{\mathrm{C}(M_i \to M_j)}{\sum_k \mathrm{C}(M_i \to M_k)}. \tag{9}$$

Once we have an approximation to Π then an estimate of the relative statistical weights of the macrostates can be made. There are a number of possible methods: the correct statistical weights will necessarily be an eigenvector of Π so a general eigenvector solver could be used. This proves problematic because the statistical weights can span a sufficiently large range that they cannot be directly represented by a double precision floating point number. It would be preferable to work with the logarithm of the statistical weight (which is implicit in the definition of η in any case). A more simplistic approach to finding the limiting distribution of a matrix is repeated multiplication: if we keep applying the same matrix to an arbitrary starting vector we will eventually converge on the correct solution. Alternatively we can repeatedly square the matrix until the solution converges along the leading diagonal. This is troublesome in practice because of the problems of numerical representation and would in any case be rather slow. It is also possible to use a general minimisation approach [73].

The preferred method for extracting the weights actually simplifies the representation a bit: we throw away all the data in the matrix except for the values on the diagonal and in the immediately adjacent rows and columns. The amount of data we lose through such a move depends on the details of the particular system under study, but is often not that significant. In any case, it is always superior to methods which do not sample

the transition probabilities at all. The weight function can be estimated from these immediately off-diagonal elements:

$$\eta_i - \eta_j = \ln \frac{P(M_i \to M_j)}{P(M_j \to M_i)}. \tag{10}$$

The full weight function follows from iteration of that rule. Errington has done a comparison of some of these techniques [73], and the repeated multiplication and diagonalisation methods are compared in chapter 6.

It would be a violation of detailed balance (not to mention computationally taxing) to recompute the weight function after every move. In practice we can update it rather infrequently (e. g. every 50,000 moves) while still producing an efficient algorithm that effectively runs under equilibrium conditions. This ability to work at equilibrium is the chief advantage of transition matrix, the calculated weight function immediately becoming accurate. The method also benefits from incorporating all the data collected so far for each estimate; there are no separated iterations, as there are in the visited states or Wang-Landau approaches.

Ensemble averages can benefit from sampling rejected moves, and this technique has been developed within a transition matrix framework [74]; it is also functionally similar to the **waste recycling** algorithm [75, 76].

The transition matrix method has many strengths: the near-equilibrium nature of the simulation, the ability to use all the data collected so far, being able to reach out into unexplored regions even where the weight function is steep, and being able to utilise data from rejected as well as accepted moves. The transition matrix should be considered a method of choice.

3.2.4 Windowing

Windowing, originally introduced alongside umbrella sampling [36, 37], can be combined with any of the above algorithms. When a large or deep weight function must be determined it is sometimes useful to be able to split the task into a series of smaller overlapping sections, or windows. Monte Carlo simulations can easily be constrained to run only within a limit range of some order parameter, and that is exactly what we do in this case. As long as practical ergodicity is maintained within the window, the results will be perfectly valid. Once each piece of the weight function has been found, it can be stitched together for continuity. The biggest advantage is that each window can potentially be run in parallel (this is an example of "embarrassing parallelism"

because the simulations do not need to communicate at all).

Chapter 4

The critical point Lennard-Jones fluid

Since extended sampling will be central to later work, this study was selected as a testbed for the methods outlined in chapter 3. It also demonstrates how the more tractable problem of liquid–vapour coexistence can be solved within an extended sampling framework. Finally, it underlines the efficacy and transparency of using a single simulation run which visits both phases for solving coexistence problems, particularly with regard to error analysis. Use is also made of a known critical point scaling relation to extrapolate coexistence properties to the thermodynamic limit.

4.1 Introduction

The Lennard-Jones system is a standard reference system in statistical mechanics and has been extensively studied (see ref. [43] for a complete phase diagram), we specifically looked at the critical point behaviour, for which their is significant prior art (refs. [77, 78, 79] for example). The system served as a testbed for some of the extended sampling techniques reviewed in chapter 3. The potential itself is given by:

$$U(r) = \begin{cases} 4\epsilon \left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6} \right] & r < 2.5\sigma \\ 0 & r \geq 2.5\sigma \end{cases} \tag{1}$$

where σ is a parameter setting the length scale of the potential and ϵ is an energy determining the magnitude of the interaction. It has been truncated at $r = 2.5\sigma$ and remains unshifted. The Lennard-Jones potential is simple enough to make computation

quick but realistic enough to display a full range of phase behaviour. Lennard-Jones has a minimum at ($r = 2^{1/6}\sigma$, $U(r) = -\epsilon$), and crosses the x-axis at $r = \sigma$. With model parameters of $\epsilon/k_B = 119.8\,\mathrm{K}$ and $\sigma = 3.41\,\text{Å}$ it also forms a reasonable model of Argon [43].

The aim of this study was to verify coexistence properties given in the literature [80], for a number of system sizes, and use a known scaling relation [81] to extrapolate these results to the thermodynamic limit. In order to use the scaling relation, we require the absolute free energy of the system as a function of system size. We chose to do this by connecting the critical point fluid to a reference state of known free energy, namely the ideal gas. A single simulation with extended sampling was used (contrast with thermodynamic integration, §2.2.1, which requires multiple simulations). In order to access this low density state, the simulation required a suitable weight function, to be obtained via some of the weight function methods discussed in chapter 3. Both the visited states and Wang-Landau algorithms were applied, for a variety of their respective adjustable parameters, to allow comparison and ensure that suitable choices could be made for the more computationally taxing studies undertaken later.

4.2 Methodology

The simulation was performed in the grand canonical ensemble (μVT) with cubic periodic boundary conditions and system cell dimensions of $L = 7.5\sigma$, $L = 10\sigma$ and $L = 12.5\sigma$. All simulations were performed at the liquid–vapour critical point, with reduced temperature $T^* \equiv kT/\epsilon = 1.1876(3)$, and reduced chemical potential $\mu^* \equiv \beta\mu = -2.778(2)$ [80].

The Hamiltonian is $\mathcal{H}(\Gamma) = \mathcal{U}(\Gamma) - \mu N(\Gamma)$, with μ the chemical potential and $N(\Gamma)$ the particle count for each microstate. The internal energy, $\mathcal{U}$, is a simple pairwise sum over particle separations. The kinetic term is omitted as per §1.5. The potential truncation ensures that each particle only interacts with the nearest periodic image of each other particle (if at all), and has no self-interaction circa the boundary conditions.

The grand canonical ensemble is particularly convenient for investigating the characteristics of fluids since it is not only computationally efficient at low densities, but also leaves the density conveniently discretised. This makes use of the particle number as an order parameter particularly straightforward. It will sometimes be useful to separate

the contributions from each discrete density from the total partition function:

$$Z = \sum_{N=0}^{\infty} \mathcal{Z}(N) \tag{2}$$

where $\mathcal{Z}(N)$ is the *constrained partition function* at a given particle number, N, i.e. represents a macrostate of the system. Assuming distinguishable particles, $\mathcal{Z}(N)$ is:

$$\mathcal{Z}(N) = \frac{1}{\Lambda^{3N}} \prod_{i=1}^{N} \left\{ \int\int\int_V d\mathbf{r}_i \right\} e^{-\beta \mathcal{H}(\mathbf{r}^N)} \tag{3}$$

where $\mathbf{r}_i$ is the ith particle's position. The constant $\Lambda \equiv h/\sqrt{2\pi m k_B T}$ (where h is Planck's constant and m the particle mass), is the thermal de Broglie wavelength and is included to make the partition function dimensionless. By convention it is taken to be unity when quoting simulation results, effectively setting the scale of the chemical potential.

The Metropolis acceptance probabilities, incorporating a multicanonical weight function, are [82]:

$$a(N \to N+1) = \min\left\{1, \frac{\eta(N)}{\eta(N+1)} \frac{V e^{\beta\mu}}{N+1} e^{-\beta \Delta U_I}\right\}, \tag{4}$$

$$a(N \to N-1) = \min\left\{1, \frac{\eta(N)}{\eta(N-1)} \frac{N e^{-\beta\mu}}{V} e^{-\beta \Delta U_D}\right\}, \tag{5}$$

where ΔU_I and ΔU_D are the potential energy changes *that would* be caused by particle insertion or deletion respectively. A good introduction to simulation in the grand canonical ensemble can be found in reference [83].

The particle number was restricted to lie between 0 and some upper bound, $N_{\max}$, by rejecting all attempted moves out of the region. The upper bound was chosen to be sufficiently large that $\mathcal{Z}(N)$ was negligible for all $N > N_{\max}$.

Trial insertions and deletions were attempted randomly, with equal probability. Observables were sampled at fixed intervals and analysed as a post-process. The energy calculation was the straight forward $\mathcal{O}(N)$ calculation as more sophisticated methods with better scaling characteristics would have been of questionable value for the relatively small system sizes studied. One optimisation was made however: 'inverting Metropolis' which is described in appendix A.

4.2.1 Weight function algorithms

The visited states algorithm was implemented as per §3.2.1, starting from a flat weight function and aiming for a flat histogram of visited states across the full density interval.

Choosing the **iteration interval**, the number of trials to attempt before an iteration is made (cf. equation 5), is critical to visited states. One weakness of the method is that it neither automates the iteration process nor suggests a suitable iteration interval. Hence a number of different lengths were tested, so that an optimal one could be determined. The trade-off is that smaller intervals result in quicker iterations, but also worse statistics, which may hinder (or even prevent) expansion into unexplored regions.

The oversampling method was used to obtain better statistics more quickly. Inevitably, there are a variety of other possible optimisations that were not investigated here. Windowing (see §3.2.4) might well have been effective for the test system. The density range could have been split into overlapping sections, each section individually converged, and the resulting pieces 'stitched together' to form the complete weight function. This refinement was not used because the aim of the study was to compare the different available methods generally, and the splitting of the histogram into sections would not necessarily be as effective in a 'typical' system or another ensemble.

The Wang-Landau algorithm was implemented as per §3.2.2, using the most common iteration rule, $f \leftarrow f/2$, and aiming for a flat histogram to within some tolerance as a precondition to iteration.

4.3 Simulation results

4.3.1 Unbiased simulation

Firstly, a number of unbiased simulations were performed for reference. Figure 4.1 shows the density histogram for the $L = 7.5\sigma$ system. Note that this is a critical system and only a single phase is present. The appearance of distinguishable peaks is a finite size effect which would disappear in the thermodynamic limit. This simulation was run for 10^{10} trials. Only about 10 % of the density range is completely unexplored, but statistics are poor for a larger proportion of macrostate. The density autocorrelation (by particle number) for the same simulation is given in figure 4.3. The radial distribution function is shown in figure 4.2. The discontinuity in the radial distribution is a result of the potential truncation used.

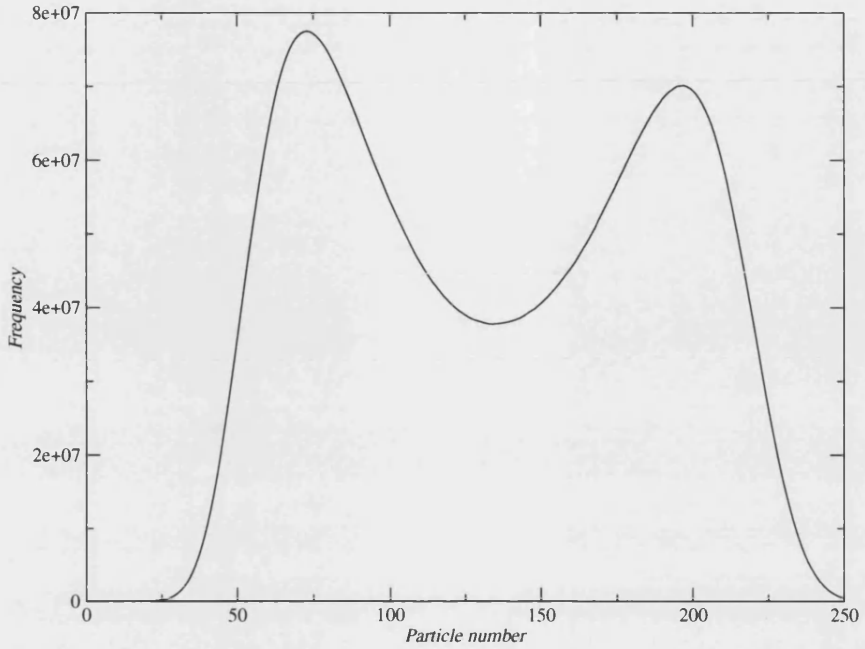


Figure 4.1: Density histogram for $L = 7.5\sigma$ system for an unbiased simulation of 10^{10} steps. Error bars would not be visible on this scale.

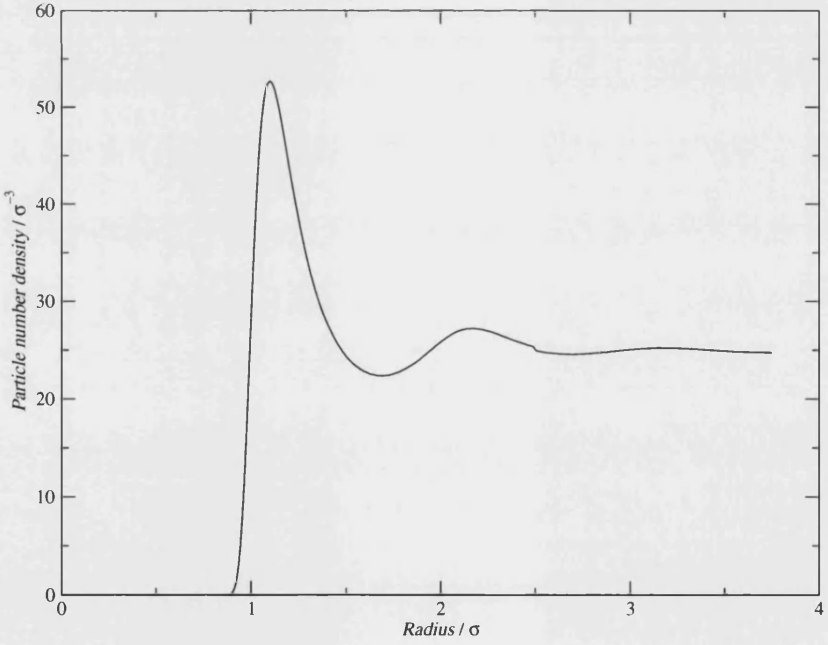


Figure 4.2: Radial distribution function for the simulation of figure 4.1.

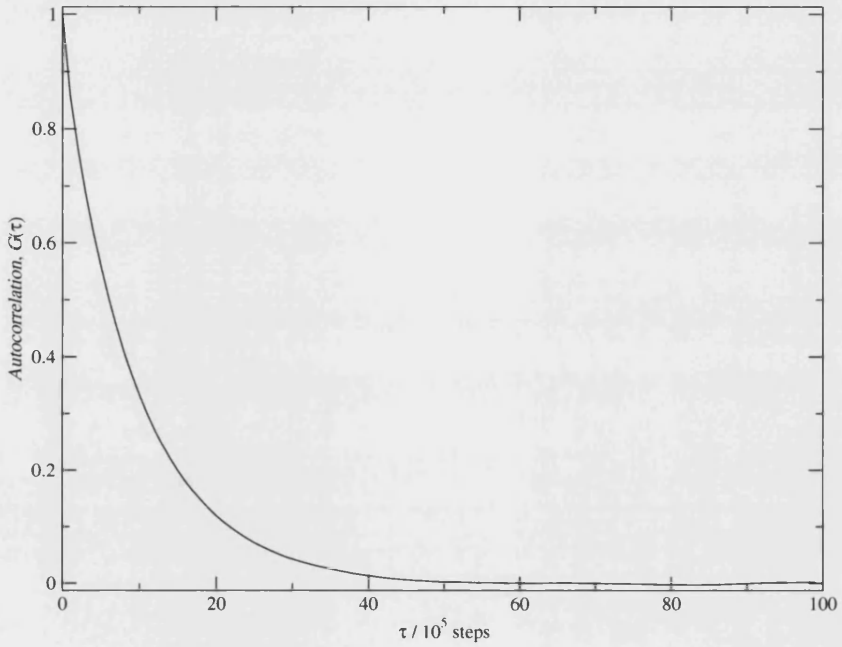


Figure 4.3: Particle number sampled every 10^5 steps and autocorrelated for system of figure 4.1. The normalised autocorrelation is calculated as $G(\tau) = \left[\langle N(t)\,N(t+\tau)\rangle - \langle N(t)\rangle^2\right] / \left[\langle N(t)^2\rangle - \langle N(t)\rangle^2\right]$, where t is MC time and τ is the delay.

Iteration interval / 10^6 trials	*Time taken* / minutes		
	$L = 7.5\sigma$	$L = 10\sigma$	$L = 12.5\sigma$
20	41.9 (15)†		
50	42.1 (6)	336 (25)†	
100	83.5 (6)	388 (14)	*
150	126 (6)	571 (14)	*
200		*	*
250		*	
300			*
400			*

Table 4.1: Total time taken for the VS simulation with given iteration and system sizes. The numbers in parenthesis are the number of iterations taken to converge. †This simulation was 'unstable'. *This simulation did not converge within 24 hours.

The visited states algorithms was then used to compute a broad histogram weight function. All the timings quoted in this section were obtained with exclusive use of an Ultra-SPARC III processor in a cluster computer and are directly comparable. The Monte Carlo trial moves took approximately twice as long for the $L = 10\sigma$ as the $L = 7.5\sigma$ system, and for the $L = 12.5\sigma$ system took about 10 times the $L = 7.5\sigma$ figure.

Critical to the convergence of visited states is the iteration interval: a low value leads to poor statistics and little progress into unexplored regions. Empirically it is observed that in extreme cases the process can break down and regions which previously had adequate statistics can go unexplored in later iterations, essentially because artificial barriers have been created by poor statistics. Such simulations will not normally converge and indicate an insufficient iteration interval. Conversely, performing unnecessarily long simulations is a waste of CPU time. The results for different intervals are shown in table 4.1.

A limit of 24 hours was imposed on the calculation, unfortunately proving insufficient for any of the $L = 12.5\sigma$ simulations to converge satisfactorily. The two simulations marked with '†' suffered from severe fluctuations hence taking many iterations, but both still eventually converged and (just) outperformed the next larger iteration interval. Use of these lengths is not recommended, and the 'optimal' values quoted later are not based on these results. Preliminary experiments confirmed that the use of oversampling was essential to obtain results in the time quoted here.

The reason the $L = 10\sigma$ system took about 10 times as long to converge as the $L = 7.5\sigma$ system (using the 'optimal' iteration interval) is the slower energy calculation, and also that visited states progresses by an approximately constant number of macrostates with

Flatness	*Time taken* / minute		
	$L = 7.5\sigma$	$L = 10\sigma$	$L = 12.5\sigma$
20 %	31.6	201	*
50 %	46.6	360	840

Table 4.2: Time taken for each Wang-Landau system size to converge, for two different values of the flatness. *This simulation did not converge in 24 hours.

each iteration, but has more ground to cover in the larger system. Also note that the algorithm makes the most progress after the initial iteration; in the smaller system it has actually already explored much of the density range just from data in the tails of the initial distribution. Larger systems gain proportionally less from the first iteration.

4.3.2 Wang-Landau

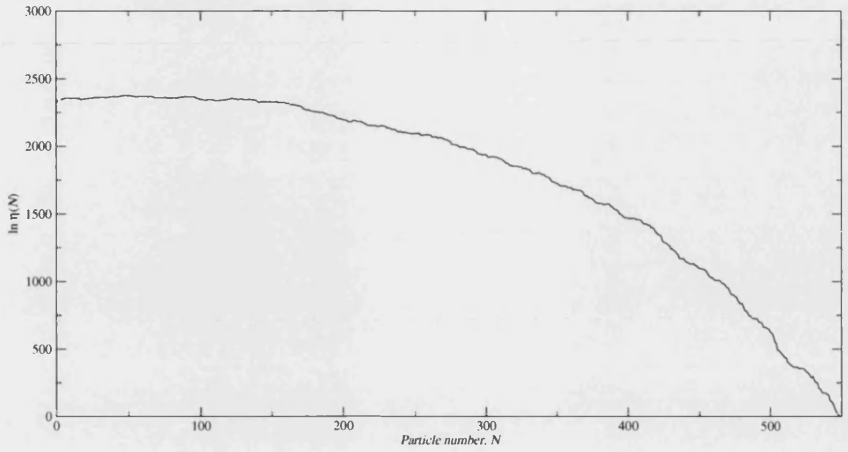
The Wang-Landau approach gave more unexpected results in places, nevertheless demonstrating its viability for a study of this kind. The first noteworthy aspect was the considerable amount by which the weight function varied as a function of f, so much so in fact, that for early iterations the equivalent statistical weight ($e^{-\eta(N)}$) is not necessarily representable as a double precision floating point number. See figure 4.4 for a typical example of the evolution of the weight function. It was also confirmed that the weight function of figure 4.4(a) did not change shape if the simulation was run for 100 times as long, thereby confirming that this is the limiting (steady state) form.

Twenty iterations were performed, terminating with $f = (1/2)^{20} = 9.54 \times 10^{-7}$ since that proved sufficient to give a weight function that sampled a flat histogram under equilibrium ($f = 0$) conditions. Both 20 % and 50 % flatness criteria were tried. To make the results more directly comparable the final iteration was altered to obtain a flatness of 20 % in all cases. See table 4.2 for the results.

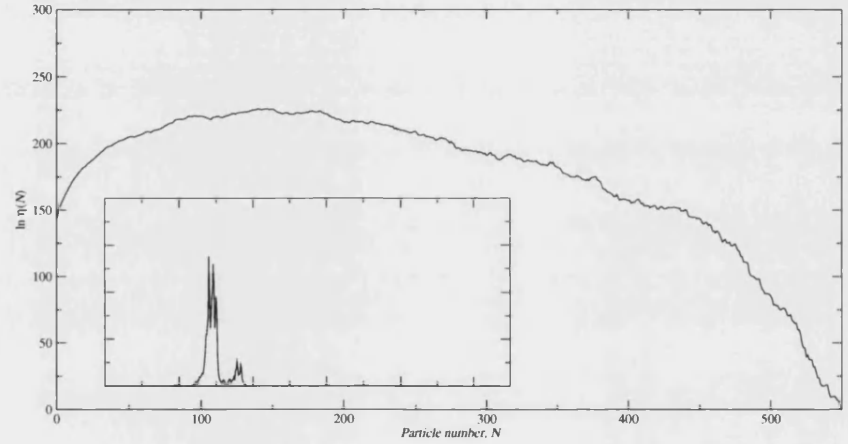
Unlike visited states, the Wang-Landau algorithm took much longer for its later iterations to converge than the early ones–a result of the exponentially smaller changes to $\eta(N)$ made in later iterations. The dependence is shown on a log scale in figure 4.5, demonstrating that much of the time was spent on the last few iterations and highlighting that careful selection of the termination point can save a considerable quantity of computation.

The 'distorted' non-equilibrium weight function obviously must be compensating for an effect introduced by the dynamic updates which modifies the effective statistical weights of the various macrostates. Since the shape of the curve changes as f is

(a)



(b)



(c)

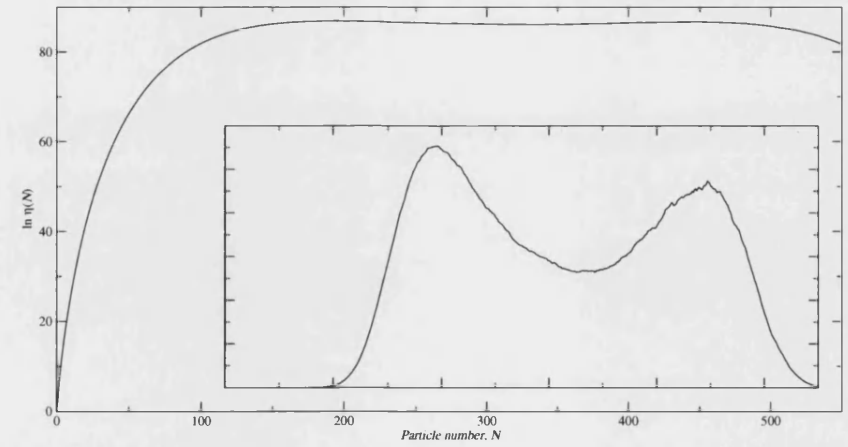


Figure 4.4: Wang-Landau weight function estimate for the $L = 10\sigma$ system with a flatness of 20 % after the (a) 1st (b) 4th (c) 20th (final) iteration. Full sized graphs depict $\ln \eta(N)$, and the insets show $\eta(N)$.

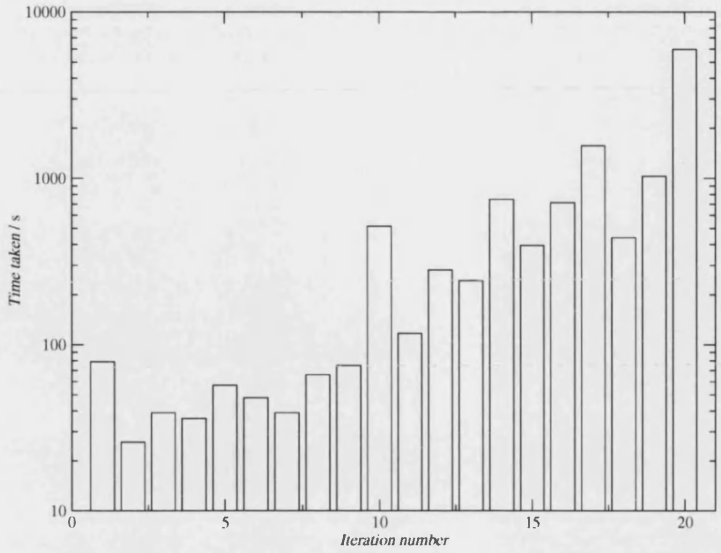


Figure 4.5: Time taken for successive iterations of the Wang-Landau algorithm for the $L = 10\sigma$ system using 20 % flatness; shown on a log scale.

decreased in a way that is asymmetric it can be concluded that it must originate from an underlying asymmetry in the system. The most likely parameter that varies with N is the acceptance probability. A qualitative argument as to the operation of this mechanism might go thus: in the high density (low acceptance) region, the estimate of the weight function 'overshoots' the value necessary to allow progression to adjacent macrostates because most moves are rejected in the steepest part of the weight function (see §3.1.3). Such 'overshooting', however, seems to suggest an overestimate in the relative weight of two adjacent N values and thus a multiplication of $e^{-\eta(N)}$ whereas the observed effect is (approximately) multiplication of $\eta(N)$ so the issue is unresolved.

Also of interest is the initial behaviour of the simulation. Proceeding from a flat weight function, the simulation tends to gradually work its way up to large N values. Figure 4.6 shows a representative example. To achieve the limiting steady-state form (see figure 4.4(a)), the simulation must spend longer at low densities initially (to fill the area under $\eta(N)$).

4.3.3 Hybrid approach

The observation that Wang-Landau has exponentially growing iteration intervals suggests that stopping at the earliest feasible juncture is critical to optimal timing. With

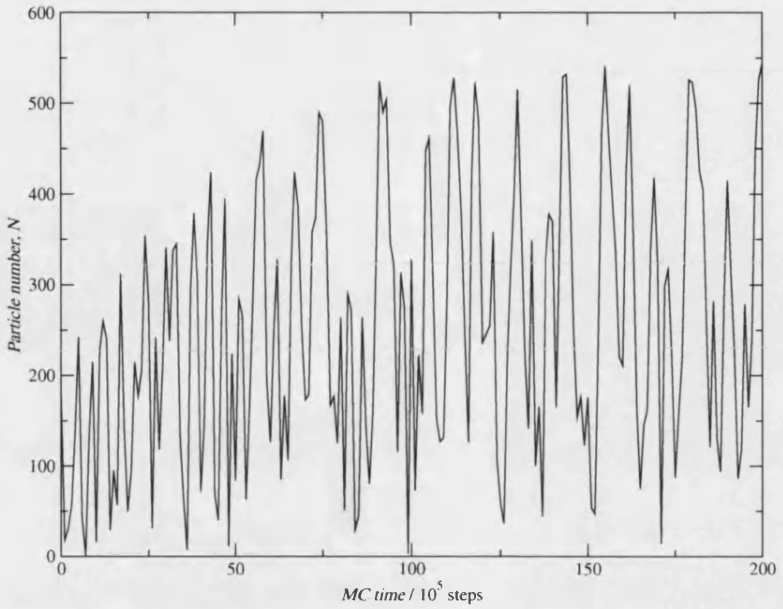


Figure 4.6: Initial behaviour of Wang-Landau simulation during the first iteration, for the $L = 10\sigma$ system.

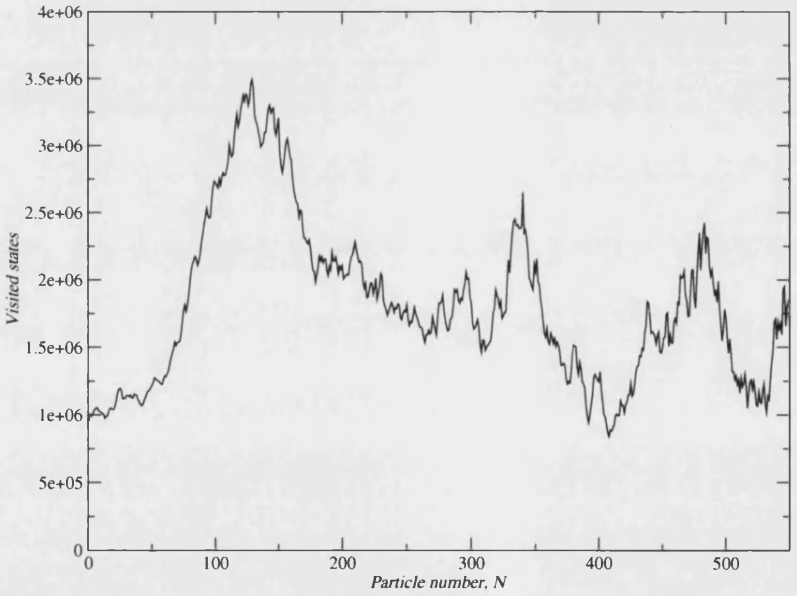


Figure 4.7: States visited by unbiased simulation using a partially converged weight function from Wang-Landau scheme after 13th iteration, for the $L = 10\sigma$ system with 20 % flatness. This simulation was 10^9 trials.

Type	*Time taken* / minute		
	$L = 7.5\sigma$	$L = 10\sigma$	$L = 12.5\sigma$
Partial Wang-Landau	4.2	20	130
Visited states iteration	11	48	410

Table 4.3: Timings for the Wang-Landau algorithm to the end of the 13th iteration, and of a single (double length) visited states iteration. Performed on $L = 10\sigma$ system with 20 % flatness for the WL part.

this in mind, a hybrid of Wang-Landau and visited states can be constructed which combines the desirable characteristics of both. The Wang-Landau algorithm is used initially, forcing exploration of the entire histogram from the outset. Once f has been reduced sufficiently that the (non-equilibrium) weight function is similar enough to the target weight function then the switch to visited states is made. As long as the weight function is not too distorted for visited states to explore the entire histogram with reasonable statistics then we can simply unfold the single visited states iteration to obtain the flat histogram weight function immediately. There is also the benefit that this was actually collected with an equilibrium ($f = 1$) simulation rather than one with $f \approx 1$ thus we know there is no out-of-equilibrium bias remaining. The difficulty in such an approach is deciding the switching point.

To test the approach, the Wang-Landau steady state results were examined manually. Interim weight function estimates from iterations corresponding to the appearance of a bimodal coexistence peak were input to a single visited states iteration. Switching after the 11th, 13th, 15th and 18th Wang-Landau iterations was attempted for the $L = 10\sigma$ system. All except the first gave an iteration that completely explored the density range (see example of switch after the 13th iteration, figure 4.7).

After it was determined that this was a good point to switch, similar procedures were carried out for the $L = 7.5\sigma$ and $L = 12.5\sigma$ systems (20% flatness variants), switching after the 13th iteration. The time taken for the Wang-Landau iterations up to this point in the respective simulations are shown in table 4.3, also shown is the time taken to do the visited states iteration, conservatively chosen to be twice the length of the 'optimal' iteration size determined previously.

Even with the additional visited states iteration, this is still about a factor of 3 times faster than the plain Wang-Landau, and around an order of magnitude faster than pure visited states.

Automating the switch-over point is desirable if many simulations are to be run. It is not necessarily obvious how this is best implemented, but a suitable approach might

System size / σ	*Free energy density / $kT\sigma^{-3}$*
7.5	-0.093356
10	-0.092493
12.5	-0.092096
∞	-0.091785†

Table 4.4: Free energy density for system sizes considered, and projection to infinite size. Error bars were not calculated for the individual system sizes. †Value calculated by extrapolation of linear fit in figure 4.8.

be: perform an unbiased simulation before the main iterative procedure, this will explore only part of the full histogram. Now use Wang-Landau as before; but after each iteration compare the statistically sound region of the initial simulation to the corresponding region of the latest estimate of the weight function. If they match within some threshold then switch to visited states. In practice this comparison could be made by normalising one of the histogram fragments to the other and checking that the value for each macrostate matches within some constant multiplier (e. g. 2).

4.4 Free energy calculation

After unfolding the weight function (see §2.1.3), an estimate of the ratio of the statistical weight in the one particle case, $\mathcal{Z}(1)$, to the total partition function, Z, can be made. Since this is a ratio of partition function like quantities it can be successfully evaluated in a Monte Carlo simulation. But $\mathcal{Z}(1) = e^{\beta\mu}/\Lambda^3$ can be evaluated analytically since the lone particle must have $\mathcal{U} = 0$ (since the potential cutoff prevents interaction with its own image). Thus the absolute (reduced) free energy ($F^* \equiv \beta F$) of the system can be calculated from:

$$F^* = -\ln\left(\mathcal{Z}(1)\frac{Z}{\mathcal{Z}(1)}\right), \tag{6}$$

where the partition function ratio is estimated from computer simulation. The calculation was partially validated by a numerical integration (not importance sampled) of the two particle state. The resulting free energy values are shown in table 4.4.

The reason the value varies with system size is that of finite-size scaling, a well documented effect that has been studied analytically at the critical point. For the studied system we expect the following scaling law [81]:

$$f_L = f_\infty + \frac{U_0}{L^3}, \tag{7}$$

where f_L is the dimensionless free energy density for system size L ($f = \ln Z/L^3$), U_0

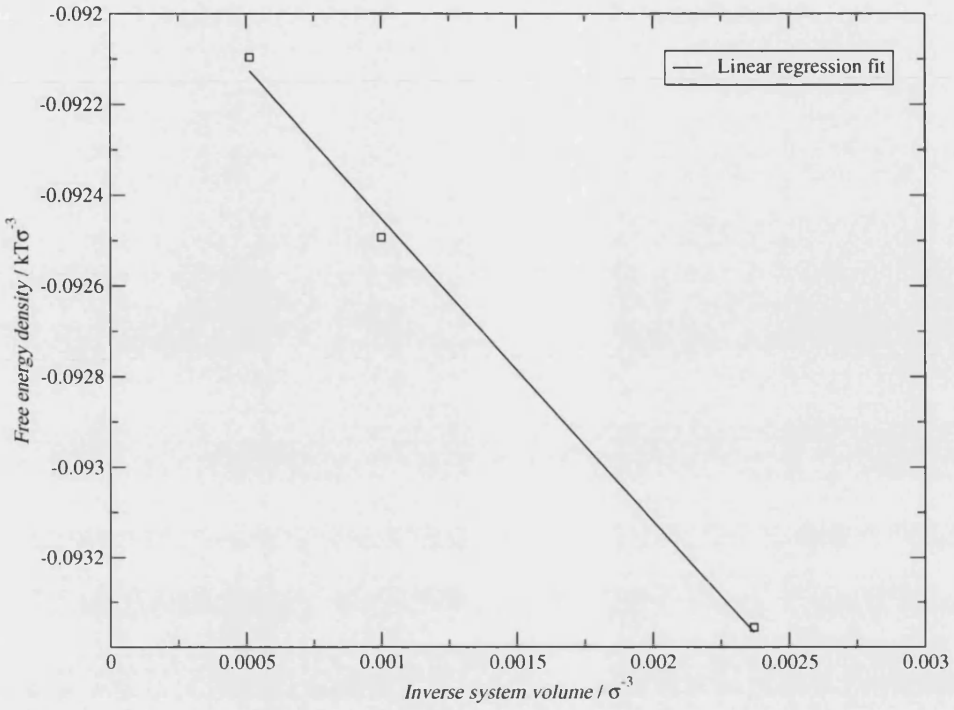


Figure 4.8: Free energy density as a function of inverse system volume. Line is a linear regression fit to the data. Error bars have not been calculated.

is the Privman-Fisher coefficient and L^3 is the system volume. U_0 for a 3D system is given as $U_0 = -0.57$ [84]. Plotting f_L against L^{-3} should therefore give a linear relationship, the gradient of which is U_0. This plot is shown in figure 4.8.

Although the available 3 points are insufficient for a precise determination of U_0, a reasonable estimate can be made. Performing a linear regression fit to the data gives $U_0 = -0.56(7)$ which is within tolerance of the value quoted above, given the lack of points. The linear fit was also used to extrapolate a value for the infinite system free energy density (shown in table 4.4).

4.5 Conclusion

It is clear from the timings presented in the previous section that, at least for the systems studied here, the Wang-Landau method significantly outperforms the visited states algorithm. Further, the hybrid approach presented above is believed to offer significant advantages over either of the pure approaches. Although implementation is

Method	*Time taken* / minute		
	$L = 7.5\sigma$	$L = 10\sigma$	$L = 12.5\sigma$
Visited states	42	390	*
Wang-Landau	32	200	840
Hybrid	11	48	410

Table 4.5: Optimal timings for each of the algorithms, for each system size. *No visited states simulation converged within 24 hours for the $L = 12.5\sigma$ system.

obviously more complicated the improvement in speed, and possibility of better scaling characteristics, may well warrant the use of such an approach in future work. The issue of automation can probably be overcome for most studies by a simple manual selection of the switch point (a potential automatic algorithm has also been presented). The main advantage is, however, that the algorithm is likely to be effective for a wide variety of problems without recourse to adjustable parameters.

The Wang-Landau approach also offers the significant advantage that it did not require a study in advance to determine the ideal iteration interval. In practice this was very time consuming and made Wang-Landau much more attractive than visited states. It is thought that the simple parameters $f_1 = 1$, $f_{\text{final}} \leq 0.00001$, $f_{n+1} = f_n/2$ and a flatness of 20 % will serve many purposes without further refinement.

The visited states algorithm is, however, better understood. In particular the violated detailed balance of Wang-Landau has not been treated analytically. But any approach that has a 'flat histogram' multi-canonical weight function as its end product will serve, and of course its correctness is trivial to test, so we do not consider this aspect a particular failure.

The optimal timings for the various algorithms are shown in table 4.5. The figures for the hybrid approach are conservative since the visited states iteration was unnecessarily long. It should be noted that only small systems were considered in this study, and larger systems may respond differently. In general we can say that the Wang-Landau algorithm out-performed visited states for this system, and is likely to do so for a range of systems.

Chapter 5

Phase switch Monte Carlo of hard spheres

This chapter introduces the phase switch Monte Carlo method, which is the principal subject of the thesis. A new code base was developed for this study, designed to facilitate the later extension to soft potentials (see chapter 6), as well as providing an independent implementation of the phase switch method for comparison with existing literature. The hard sphere system was chosen for program validation against known results. Since the extended sampling problem is more severe than in chapter 4 it was necessary to use a more sophisticated framework for deriving the weight functions.

5.1 Review

The problems of phase equilibria amongst crystalline solids (CS), and between fluids and CS have traditionally proved difficult [85]. The earliest attempts for hard spheres, itself the prototypical system, date back almost 40 years to the work of Hoover and Ree [94, 95] and revolved around the use of a **single occupancy cell model**, later extended by Hoover *et al* [96]. Thermodynamic integration (see §2.2.1) was performed along an artificial route: the particles were constrained to cells surrounding their lattice sites, and the cells were then inflated until the crystal had effectively melted. Connection could then be made to a low density reference state (ideal gas). Frenkel and Ladd instead chose to link the hard sphere solid to the Einstein crystal, for which the free energy is known analytically [97]. Cowley and Barker attempted an analytical treatment based on a cell model for both phases [98]. Later, a density functional treatment was developed [99, 100]. Recently, a study of hard sphere freezing in very small isolated

clusters was performed [101]. For such small systems, freezing *does* occur on simulation timescales, however the results cannot be extrapolated to bulk properties. Finally, hard sphere freezing has also been investigated experimentally [102].

What all the computational methods share is a reliance on unconstrained approximations, or on the integrity of thermodynamic integration along a tortuous inter-phase path. Ideally, we would like to produce a single Monte Carlo simulation which takes us directly through the transition, as is possible for the liquid–vapour transition. Such a single simulation allows us to directly compare the relative stability of the two phases as a ratio of single-phase partition functions. The most obvious benefit is transparent error calculation; techniques such as Gibbs-Duhem integration and thermodynamic integration can potentially accumulate systematic errors, whereas within a single simulation all errors are statistical given the usual provisos common to all Monte Carlo simulations such as practical ergodicity and the quality of the pseudo-random numbers.

The first simulation method to connect distinct crystalline phases was that of lattice switch Monte Carlo [85, 90], later extended to soft potentials [103]. Whereas extended sampling catalyses phase transitions without inherent symmetry (liquid–vapour, liquid–liquid and glassy), lattice switch, by contrast, does the same for solid–solid transitions. Finally, phase switch Monte Carlo [104, 105], the subject of this chapter, generalises lattice switch to fluid–solid transitions.

5.2 Statistical mechanics

For hard spheres of diameter D, the potential is simply:

$$U(r) = \begin{cases} 0 & r > D \\ \infty & r \leq D \end{cases} . \tag{1}$$

The Hamiltonian of a system of hard spheres in the isothermal–isobaric (NPT) ensemble may be written:

$$\mathcal{H}(\Gamma) = \mathcal{U}(\Gamma) + P\,V(\Gamma) \tag{2}$$

where $\mathcal{U}$ is the internal energy, i.e. infinite for configurations containing any hard sphere overlaps or zero for ones that do not.

Equation 2 implies a sampling distribution of:

$$\mathcal{P}(\Gamma) = \frac{1}{Z} e^{-\beta PV(\Gamma) + \beta \mathcal{U}(\Gamma) - \eta(\Gamma)}, \tag{3}$$

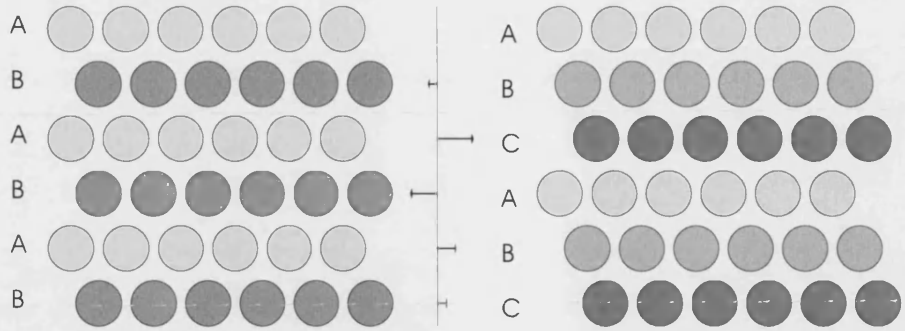


Figure 5.1: The FCC (left) and HCP (right) phases for hard spheres. FCC can be thought of as alternation between two layers, while HCP cycles through three. The lattice switch displacements are shown in the centre, which when applied to each layer, would transform the FCC crystal into the HCP one.

showing that β does not influence the internal energy term. As such the behaviour of a hard sphere system can be characterised by a **reduced pressure**, $P^* \equiv \beta P$. There is effectively only one external control parameter and the phase diagram becomes entirely one dimensional. The partition function follows as:

$$Z = \int_0^\infty \mathrm{d}V \prod_{i=1}^{N} \left\{ \iiint_V \mathrm{d}\mathbf{r}_i \right\} e^{-\beta\mathcal{H}(\Gamma)-\eta(\Gamma)}. \tag{4}$$

5.3 Lattice switch Monte Carlo

Lattice switch operates by introducing a novel Monte Carlo move: a global translation designed to instantaneously move directly from one phase to the other without passing through interfacial states. It is sometimes termed a **phase leap** or **teleport** move. We shall examine the particular case of the FCC–HCP transition for hard spheres [85], the first application of lattice switch. The HCP phase can be broken down into a series of parallel layers running through the crystal. There are three types of layers, termed the 'A', 'B' and 'C' layers which cycle through the crystal in that order. Conversely, the FCC phase has only alternating 'A' and 'B' layers. Figure 5.1 shows a two dimensional analogue of the arrangement.

The insight lattice switch trades on is that we can create a mapping between the two phases by performing a series of horizontal (with respect to the layers as we have identified them) translations which serve to rearrange one crystal structure into the other. Each layer is moved as one unit. Since the least common multiple of 2 and 3 is 6, the pattern of displacements required will repeat with that frequency.

The lattice switch is fully reversible since the inverse displacements will effect the opposite phase transition. The lattice switch moves can be attempted throughout the simulation, generating many crossings and hence good statistics. One of the most attractive features of this methodology is that we do not have to leave the pure phase because the link is not created by forcing the system through an artificial inter-phase path (as in constrained fluid λ-integration, cf. §2.2.5).

The problem with the method as it stands is that because the particles fluctuate around their rest positions, it is very likely that any given lattice switch move would result in hard sphere overlaps. We overcome this by appealing once again to extended sampling. We define an order parameter, M, expressly for the purpose of driving the simulation into a state where it can successfully change phase. We track the state *that would* result from an immediate lattice switch at all times, except that we replace the hard sphere potential with a soft potential (a ramp typically) in this virtual phase. At every normal Monte Carlo move we also calculate a 'conjugate energy' function as a sum of these ramp potential terms for the alternate phase. If this virtual energy function can be reduced to zero, then we have cleared the way for a successful lattice switch. Extended sampling can now step in and bias the simulation towards these low 'conjugate energy' states. If we aim for a flat histogram in M then the simulation will both visit the equilibrium states (typically with high M) as well as those of low M from which a lattice switch is possible. We refer to these states where $M = 0$ as **gateway states**.

Another weight function will be required in the other phase in order to link that phase to its corresponding gateway states. In the case of hard spheres we can simply match the gateway states in one phase to have the same multicanonical weight as the gateway states in the other, in which case a lattice switch attempt from a gateway state will always succeed. By construction, the gateway states must have equal phase space volume and, therefore, statistical weight in each phase.

5.4 Phase switch Monte Carlo

Similarly to lattice switch, **phase switch Monte Carlo** aims to directly link the two phases via a phase leap. The benefits of such a transformation are plain; the interfacial states are circumnavigated. The formalism that permits this move is developed in this section.

Firstly, all the particle positions ($\mathbf{r}_i$) are represented as displacements ($\mathbf{u}_i$) from posi-

tions on a reference lattice, $\mathbf{R}_i$:

$$\mathbf{r}_i = \mathbf{R}_i^\gamma + \mathbf{u}_i \quad \forall i. \tag{5}$$

Two such references lattices, one for each phase, denoted $\mathbf{R}_i^{\mathrm{CS}}$ for the crystalline solid phase, and $\mathbf{R}_i^{\mathrm{F}}$ for the fluid phase are selected. These should be 'typical' configurations from each phase (i.e. clearly belong to the phase in question). In practice the reference lattices are merely randomly selected microstates from an unweighted, equilibrated simulation run in each phase.

During the simulation the 'current' phase, γ (either 'F' or 'CS'), is tracked and the u-vectors are interpreted relative to that phase's reference lattice, $\mathbf{R}^\gamma$. Now a correspondence between the two phases is proposed, namely that every microstate has a 'conjugate phase' microstate associated with it, produced by applying the current displacements ($\mathbf{u}^N$) to the conjugate phase lattice rather than the current phase one. In general this conjugate phase microstate will not be physical (will contain hard sphere overlaps). However, if the simulation happens to visit a microstate very similar to the current phase reference lattice, that is, where all the u-vectors are sufficiently small, then the conjugate phase microstate will be both physical and belong to the conjugate phase. From such '*gateway states*' a phase switch global move, where the phase label and reference lattice are exchanged, could succeed. Essentially, a one-to-one mapping between two regions has been created, bridging the two phases. In contrast to lattice switch which has a relatively general transformation, here the mapping is intensely specific: only microstates close to a *specific* reference microstate afford a phase leap. The mapping is demonstrated schematically in figure 5.2.

Naturally the probability of an unbiased simulation visiting the vicinity of the particular reference state chosen is negligible. Note there is nothing special about the reference microstate: it is just randomly selected, but the probability of visiting (the vicinity of) *any* one configuration is vanishingly small. Therefore it is necessary to perform extended sampling to bias the simulation to visit the gateway states, analogous to the lattice switch case. For this purpose an order parameter, M, is defined whose value represents how 'close' a given microstate is to the gateway macrostate. For the hard-sphere system the following choice was made [106]:

$$M = \sum_{i=1}^{N} m_i; \quad \text{where} \quad m_i = \begin{cases} \sum_{\substack{j=1 \\ j\neq i}}^{N} \max\{1 - \overline{r}_{ij},\, 0\} & |\mathbf{u}_i| \leq r_c \\ \alpha |\mathbf{u}_i| & |\mathbf{u}_i| > r_c \end{cases} \tag{6}$$

Here $\overline{r}_{ij}$ refers to the separation of the centres of particles i and j in the *conjugate phase* (circa the boundary conditions); i.e. it measures the 'degree of overlap' *that would* be

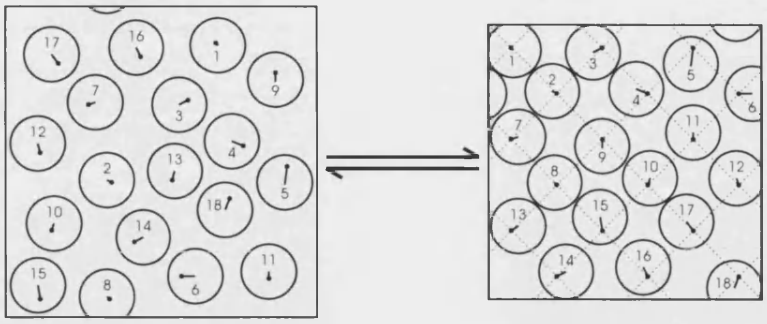


Figure 5.2: Schematic illustration of the phase switch mapping. The dots represent the reference lattice in each phase, with the u-vectors connecting the particles to their corresponding lattice site. The same u-vector is applied in each phase, but leads to a different configuration because they are interpreted relative to a different reference lattice.

created by a phase switch. The order parameter acts on a particle-by-particle basis, providing a simple proportional bias against particles far from their lattice sites, known as **tether mode**. The particles switch to a bias against conjugate phase overlaps as the particles come within a **cutoff radius**, r_c, of their associated reference lattice sites. This is known as **overlap mode**. A phase switch can only be successful from one of the gateway states, i.e. when $M = 0$. The parameter α is chosen (empirically) to make to transition between the two modes proceed without producing a discontinuity in the order parameter *on average*.

The efficiency of the phase switch is also improved by coupling the change in lattice to a volume scaling. The volume ratio of the two microstates chosen as reference lattices is used to scale the conjugate phase microstate; so that, in general, the conjugate phase microstate has a volume typical of the conjugate phase, rather than the current phase. Only the reference lattices are dilated during the volume move, the displacements remain unscaled.

5.5 Phase space fragments in crystalline solids

There is a caveat to the above discussion in that generally a simulation initiated in a crystalline phase not only suffers a large free energy barrier to crossing into other phases, but it is also unlikely that particles will travel away from their initial lattice

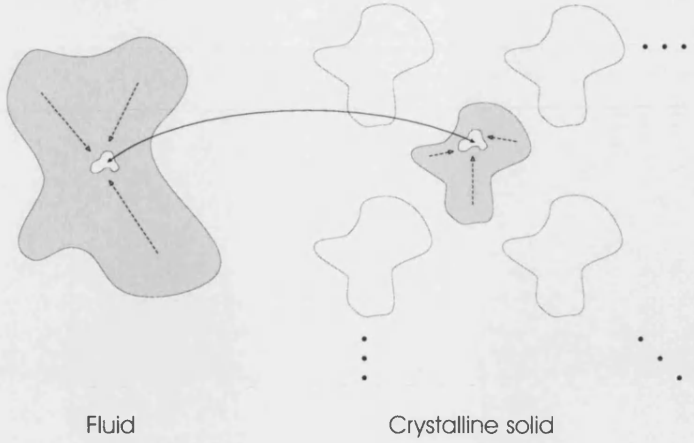


Figure 5.3: Schematic of the phase space of the problem. The crystal has many disjoint fragments, corresponding to permutations of the particles amongst the lattice sites. The phase switch maps a very small region of each phase to the other. The dotted arrows represent the bias drawing the simulation towards the gateway states.

site because of the self-constraining nature of crystalline solids. Empirically it was found that for the hard-sphere system no such changes occurred, meaning that the permutation of the particles amongst the lattice sites remained fixed throughout the simulation. For the Lennard-Jones potential (and presumably other soft potentials) such swaps did happen, albeit extremely infrequently.

A simulation that sampled only one permutation of the crystal would dramatically underestimate the statistical weight of that phase. The phase space of the crystal is divided into a number of disjoint **fragments** that are physically identical (at least in the monodisperse case), differing only in the particle labels. By symmetry these fragments all contribute equally to the total statistical weight. Figure 5.3 represents the situation diagrammatically. We choose to only let the simulation visit a single permutation of the crystal, which saves us from having to solve the diffusion problem in the crystal phase.

Unless the centre of mass of the system has been conserved somehow, the lattice as a whole will still be free to drift with respect to the simulation cell, potentially counting other permutations, but also providing an unnecessary impediment to the phase switch move itself. Thus we choose to fix an arbitrary particle thereby pinning the crystal in place [85].

Without the modification, the system is translationally invariant; i.e. for any displace-

ment $\mathbf{s}$, the transformation:

$$\mathbf{r}_i \mapsto \mathbf{r}_i + \mathbf{s} \quad \forall i \tag{7}$$

applied to a microstate cannot change the value of any observable. This property is exploited by the pinning process. Evaluating $e^{-\beta\mathcal{U}(\Gamma)}$ in terms of a set of translated coordinates, $\mathbf{q}_i \equiv \mathbf{r}_i - \mathbf{r}_1$, it is possible to re-write equation 4 in the following form:

$$Z = \int_0^\infty dV \iiint_V d\mathbf{r}_1 \prod_{i=2}^{N} \left\{ \iiint_V d\mathbf{q}_i \right\} e^{-\beta\mathcal{H}(\Gamma')}, \tag{8}$$

where the microstate Γ' is simply Γ but with the transformation applied. Now the integration over particle 1's coordinates is independent of $e^{-\beta\mathcal{H}(\Gamma')}$, so can be integrated analytically, yielding simply V. The partition function can now be re-written as:

$$Z = \int_0^\infty dV \prod_{i=2}^{N} \left\{ \iiint_V d\mathbf{q}_i \right\} V e^{-\beta\mathcal{H}(\Gamma')} \tag{9}$$

where it is understood that when evaluating $e^{-\beta\mathcal{U}(\Gamma')}$ that $\mathbf{q}_1 = 0$ always. Comparison of equations 3, 4 and 9 shows that simulating with an effective probability distribution of:

$$\mathcal{P}(\Gamma) = \frac{1}{Z} V e^{-\beta\mathcal{H}(\Gamma')} \tag{10}$$

but never moving particle 1 will give identical results to the full formalism of the previous section. This alteration is assumed in all subsequent sections, but the primes are omitted. Thus we can safely fix one particle to pin the crystal. Note that under the new partition function the full statistical weight of the crystalline phase is $(N-1)!$ times the simulation value as the fixed particle no longer enters into permutations.

5.5.1 Association swaps

In crystalline phases the particles will not, under normal conditions, succeed in swapping lattice sites with neighbours because of the constraint applied by the crystal itself. This does not follow in the fluid case, and particles may be found far from their reference lattice sites. Often one particle may actually be nearer to the reference site of another particle, and vice-versa. In this case the efficiency of the simulation can be improved by allowing a so-called **association swap**, where two particles are re-associated

with each others' reference sites, without altering the (current phase) configuration:

$$\mathbf{u}_i' = \mathbf{L}_j^\gamma - \mathbf{L}_i^\gamma + \mathbf{u}_i \tag{11}$$

$$\mathbf{u}_j' = \mathbf{L}_i^\gamma - \mathbf{L}_j^\gamma + \mathbf{u}_j \tag{12}$$

$$\tag{13}$$

which follows from $\mathbf{L}_i^\gamma + \mathbf{u}_i = \mathbf{L}_j^\gamma + \mathbf{u}_j'$. Such a move will simply produce a permutation of the original microstate in the current phase, but will significantly perturb the conjugate phase microstate. There are $(N-1)!$ possible permutations of particles, so overcoming this entropic barrier is paramount to an efficient simulation. Association swaps provide that function by permuting the particles directly. Also note that association swaps have the ability to make non-local changes in the conjugate microstate.

5.6 Simulation method

An unbiased equilibrium simulation was run for each phase, and reference lattice microstates were saved after equilibration. All simulations were performed with $N = 108$ particles, at a reduced pressure of $P^* = 11.5\ D^{-3}$ (where D is the hard sphere diameter), which is known to be in the vicinity of coexistence [106]. In the experiments, a simulation sweep consisted of:-

- $N - 1$ particle translation attempts (particle 1 was excluded, see section 5.5).
- N association swap attempts (automatically rejected in solid phase)
- A volume move attempt
- A phase–switch attempt (if $M = 0$)

The particle translation attempts were chosen uniformly up to a limit of $\Delta x_{\max} = 0.05\ D$ along each axis. Volume moves were chosen uniformly up to a limit of $\Delta V_{\max} = 0.25\ D^3$. These values were chosen to give an acceptance of around 40 %. In line with previous work [85], the parameter α in the order parameter definition (equation 6) was taken to be 1.7, and the cutoff radius, r_c to be 0.5 D. Observables were sampled every 100 sweeps.

Once the weight function had been determined as specified in section 5.7, then the phase–switch move was enabled and preweighted simulations were run to measure the actual free energy difference between the two phases. Nine separate simulations were

run, the data was collated, and the weight function was unfolded. Next the coexistence pressure was determined by histogram reweighting [82] and a simple root-finding algorithm (being careful to correct for the factor of $(N-1)!$ mentioned in section 5.5).

5.6.1 Acceptance rules

The Metropolis acceptance probabilities were derived from detailed balance and the sampling distribution (equation 3) in the usual manner [86]. The derivations of the volume move is presented in appendix B, and that of the phase–switch move in appendix C. The derivation of the probabilities for the translation and association swaps are trivial. The probabilities for all four varieties of move are:

$$
\begin{aligned}
&\text{Particle translations} && a(\Gamma \to \Upsilon) = \min\left\{1,\ e^{\eta(M')-\eta(M)}\,\delta(\Upsilon)\right\} \\
&\text{Volume moves} && a(\Gamma \to \Upsilon) = \min\left\{1,\ e^{\eta(M')-\eta(M)}\left[\frac{V'}{V}\right]^{N}\delta(\Upsilon)\,e^{-P^*[V'-V]}\right\} \\
&\text{Association swaps} && a(\Gamma \to \Upsilon) = \min\left\{1,\ e^{\eta(M')-\eta(M)}\right\} \\
&\text{Phase–switch} && a(\Gamma \to \Upsilon) = \min\left\{1,\ \left[\frac{V'}{V}\right]^{2}\delta(\Upsilon)\,e^{-P^*[V'-V]}\right\}
\end{aligned}
\tag{14}
$$

Here $\delta(\Gamma) = e^{-\beta\mathcal{U}(\Gamma)}$, i.e. unity for physical microstates and zero for unphysical ones. It has been assumed that $\delta(\Gamma) = 1$ since unphysical microstates are not visited in the first place. Particle translation trials were chosen uniformly over a cube centred on the particle origin. Volume trials were chosen uniformly over an interval in *volume* centred on the current value. The volume of the conjugate phase was always chosen as described in section 5.4.

5.7 Determining the weight functions

Computationally speaking, much of the work in implementing the phase–switch method is finding a weight function that correctly forces the system to visit the gateway states. Empirically it is observed that the required multi-canonical weight function, even for a 108 particle system, spans a huge range. Following the conclusions of chapter 4, the Wang-Landau algorithm was used initially.

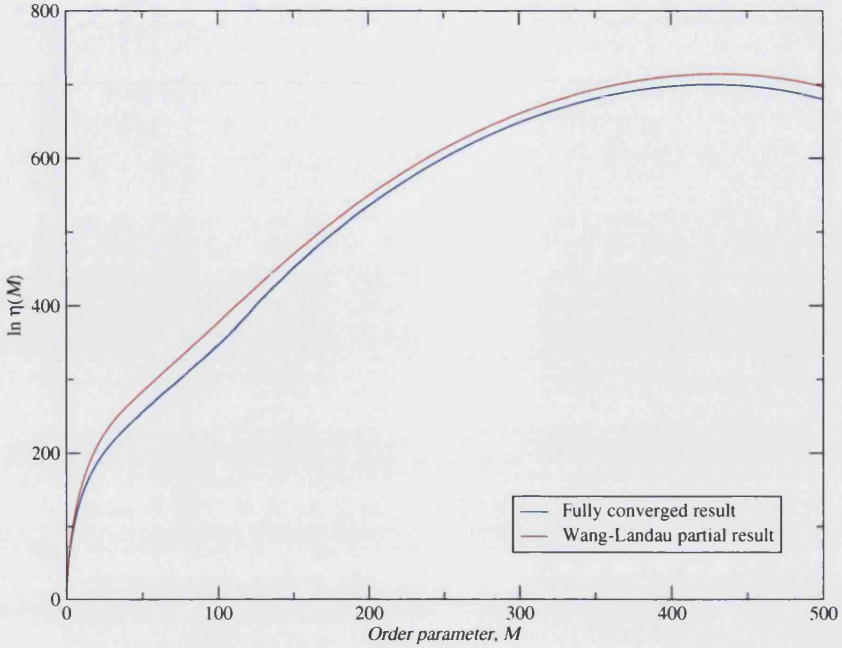


Figure 5.4: Partially converged Wang-Landau solution for the fluid phase during the 16th iteration (upper line) with $f = 1.0000457$. The solution was matched to the 'textbook' answer (lower line) by arbitrarily matching the graphs such that $\eta(0) = \eta_{\mathrm{WL}}(0) = 0$.

Determination of the weight functions was performed independently for each phase, and the phase–switch move itself was disabled throughout. The 108 particle system ($3\times3\times3$ FCC unit cells for the crystalline phase) was used for all this work.

5.7.1 Wang-Landau

The Wang-Landau approach seems well suited for determining the weight functions in this system. The ability to overcome large energy barriers, following from the 'brute force' approach to barrier penetration, together with the generality of the initial parameters make it appear attractive. Implementation was straight-forward in the phase–switch code. The algorithm used the usual $f = 1$ start condition and the $f \leftarrow f/2$ iteration rule, together with a flatness of 50 % as the iteration condition. The fluid phase was divided into 8192 ($= 2^{13}$) bins spanning the empirically determined order parameter range.

This method worked, if slowly, for the FCC phase. But, after some experimentation,

it proved to be computationally intractable for the fluid phase. This was a result of a much steeper weight function than in the previous work. The 'exaggeration' of the weight function (see section 4.3.2) was present, and much more pronounced than in the grand canonical ensemble case, leading to a prohibitively long first iteration (since the first iteration must, effectively, fill in the total area under $\eta(M)$). The problem was eased by increasing the initial value of f to 20. With this modification the simulation did proceed past the first iteration eventually; but after several weeks of computer time the algorithm had failed to converged satisfactorily. Figure 5.4 shows the partially converged solution after about 2 weeks of computer time compared against the 'book answer' as determined in later work. The Wang-Landau solution is still incorrect by a factor of approximately e^{15} ($\sim 10^7$) for the statistical weight in places. The solution within a factor of about 10 was required before visited states could be usefully employed. This approach was abandoned because of the computational expense; especially considering this was only for $N = 108$ particles and it was hoped the work would extend to larger systems.

5.7.2 Successive umbrella sampling

The observation that the difficulties with the Wang-Landau algorithm derive from the 'exaggeration' effect encourages approaches immune to this. Thus a variant of the visited states algorithm known as **successive umbrella sampling** [87] was tried. For this approach, the order parameter domain was again divided into many narrow macrostate (of width $\Delta M = 0.0610$ for the example system). The strength of the visited states algorithm is that if it visits a given macrostate a reasonable number of times, then the ratio of that macrostate's weight to its immediate neighbours will be determined with good accuracy (provided the macrostate are sufficiently narrow). This feature is exploited to generate a crude estimate of the weight function.

The simulation is performed as usual, and with a flat preweight function. At regular intervals the simulation is checked to determine if there are any macrostate that the simulation has visited more than a threshold number of times, as well as having both neighbours similarly visited more than the same threshold. For any such macrostate a snapshot of the weight ratios between the macrostate and its neighbours is saved; and then the macrostate is 'blocked'. A macrostate is 'blocked' by rejecting all moves into the macrostate (effectively setting $\eta = \infty$). This blocking effect forces the simulation away from regions it has already explored, much like the Wang-Landau algorithm. The algorithm also conforms to detailed balance.

Some implementation details are needed in addition to the prescription above. Firstly,

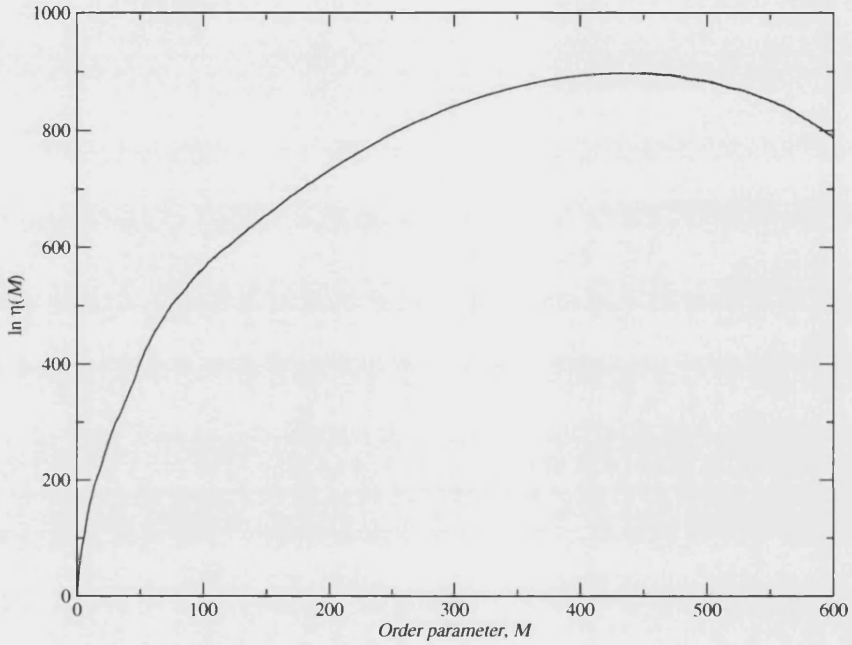


Figure 5.5: Weight function estimate produced by 'successive umbrella sampling', representing approximately 45 minutes of computer time. The threshold was set at 1000 visits per macrostate.

isolated unblocked macrostates cannot be allowed to form since they will (probably) never be visited again. Secondly, the simulation must be able to traverse any barriers (regions of blocked macrostates) that form. This is implemented by saving a random configuration belonging to each macrostate, then when the simulation completes one region it can jump back to a random 'remembered' configuration and continue.

The algorithm can very quickly produce an estimate of the weight function, but is not generally very accurate. It does operate with impressive rapidity, see figure 5.7.2 for an example. With a higher threshold (10,000 visits per macrostate) the algorithm produced a good estimate, but nevertheless insufficient to be useful (a simulation preweighted with the estimate freely explored the interval from $M = 0$ up to about $M = 10$). The principle flaw in this algorithm is the inability to make it iterative, and thus improve on the estimate. The consequences of simulating very close to blocked macrostate is unclear, and there may be subtle boundary effects which are not understood. The algorithm also has the undesirable quality that since it chains the ratios of macrostate weights together, the error between each adjacent pair of macrostate affects the entire order parameter range, a behaviour with unfavourable scaling characteristics.

5.7.3 Grouped Wang-Landau

Following the shortcomings of the Wang-Landau implementation of section 5.7.1, a variant was tried. The difficulty for very "steep" systems, such as this, is that the simulation must visit the equilibrium part of the order parameter range so many more times than the outlying regions to build up a sufficiently steep weight function. This variant seeks to reduce the problem.

The 'grouped' Wang-Landau method divides the order parameter range into a small number of groups of adjacent macrostates (16 for this study). The flatness criteria is then applied individually to each group, and once a group at the edge of the explored M interval has converged then that piece of the weight function is saved, and the simulation is prevented from returning to that group. Determining $\eta(M)$ like this allows the simulation to quickly exclude the comparatively flat portions of the weight function, and thus saves the 'maintenance' on such sections that must otherwise occur while the steeper groups are converging.

To implement this method properly it is necessary to refine the idea a little. The simulation only 'blocks' a group once not only it, but also the adjacent groups had converged. This permits the system to properly 'stitch' the weight function fragments together. It also reduces the danger of subtle boundary effects from the 'wall' (last

blocked group) on the results. The procedure is repeated for every Wang-Landau iteration (although the principle advantage comes during the first iteration).

The method proved quite effective at eliminating some of the unnecessary work that Wang-Landau performs. However, the computational sluggishness returned as the simulation entered the steepest regions of the weight function. The implementation proved ineffective for later iterations because it only allowed the barrier to travel from high M to low M values, whereas the converse behaviour would have proved more useful. This refinement could have saved some more time, but in general this method was inherently time-consuming.

5.7.4 Windowed Wang-Landau

The precept for this approach was that if the workload could be divided up then the various parts could be calculated in parallel. To this end the order parameter range was divided into four overlapping *regions*. Each region was calculated completely independently using Wang-Landau. Once the weight function fragments were determined they were matched together using the overlaps. The regions were made narrower near $M = 0$, where the weight function was known to be steepest, to split the computational cost more evenly. The four simulations covered: $M = 0$–20, 10–50, 40–210 and $200-500$. The (computationally easier) solid phase weight function was determined as a single region. Additionally a simulation with much smaller range was performed over the interval $0-5$ for each phase after it became apparent that greater precision was required in this (very steep) region. This method did produce converged weight functions within a reasonable time-frame.

5.7.5 Finalising the weight function

Once the Wang-Landau estimates of the weight function pieces had been generated they were stitched together using the overlapping parts. The combined weight function, despite successfully allowing a preweighted simulation to explore nearly the full order parameter range, proved inadequate for actually reaching the $M = 0$ gateway states themselves. The problem can be traced to the form of the weight function. The Wang-Landau method leads naturally to representation as a 'staircase' function reflecting the division into bins. This, however, fails to provide the gradient needed to encourage moves to M values smaller than the centre of the lowest bin (however narrow it is made).

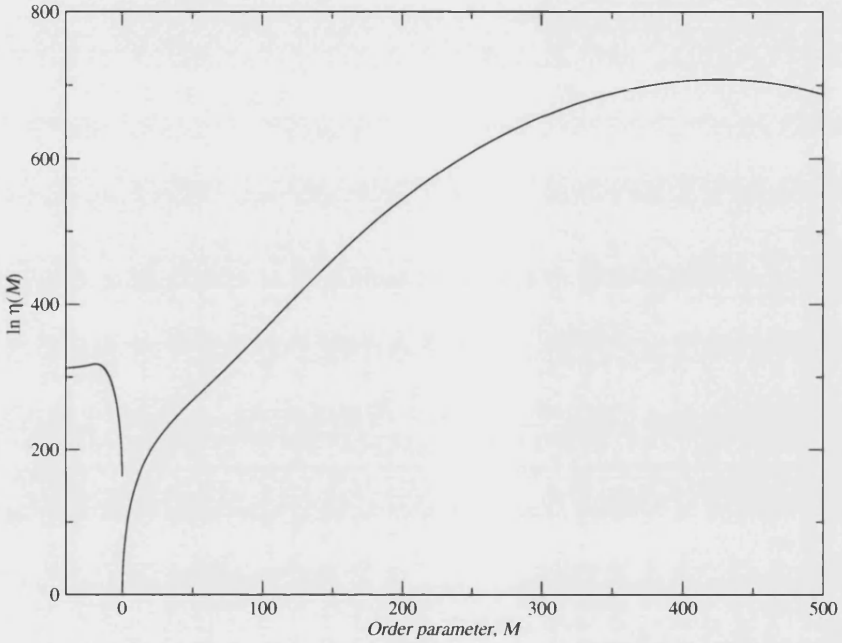


Figure 5.6: Final weight function; this is a polynomial fit to the 'Windowed Wang-Landau' data, for both phases. The crystalline solid phase is shown as negative M.

Performing linear interpolation between the bins (and extrapolation at the ends) did not prove sufficient to force the simulation to $M = 0$. Accordingly, a piece-wise polynomial fit of the data was made, using 5th and 6th order polynomials. Six pieces were used for the fluid phase and four for the crystalline phase. In both phases, the last section (abutting $M = 0$) was instead approximated by the function:

$$\eta(M) = aM^b + cM + d \qquad (0 < b < 1). \tag{15}$$

All fits were performed in `xmgrace-5.1.10`, and were matched together simply for continuity. It proved necessary to further tweak the value of the exponent b to make the function slightly steeper. With all these modifications the simulation did reach the gateway states as required. The weight difference between the two phases ($\eta_{\mathrm{CS}}(0) - \eta_{\mathrm{F}}(0)$) was selected so that, empirically, the two phases were visited with equal frequency (at least at $M = 0$). The final fitted weight functions for both phases are shown in figure 5.6, where the convention that the crystalline solid phase is associated with a negative M value has been adopted for convenience.

The reason the equilibrium sections of the two phases have such apparently disparate weights despite the simulation supposedly being near coexistence can be traced to a difference between the behaviour of the two phases. Recall that the crystalline phase cannot explore other permutations than that of the reference lattice whereas the fluid phase explores its phase volume fully (in principle). The true weight of crystalline phase will thus be underestimated by a factor of $(N-1)!$, being the number of permutations of $(N-1)$ movable particles amongst the lattice sites (noting that particle 1 never moves).

5.8 Results

A typical simulation run is depicted in figure 5.7, showing the evolution of the order parameter and density over the run. The simulation is seen to successfully visit both phases. One problem was encountered with the simulations, demonstrated by the last section of figure 5.7, which after investigation proved to be the simulation spontaneously freezing. This means that the simulation actually managed to freeze without the assistance of the phase–switch, producing misleading data. Attempts to prevent this undesirable effect by excluding volume moves leading to simulation densities greater than a cutoff (only in the 'fluid' phase) proved fruitless, because the density peaks for both phases overlapped. It should be noted that this effect would not occur in larger systems because, as previously mentioned, such transitions become prohibitively unlikely as the system size increases. The problem was overcome by manually examining

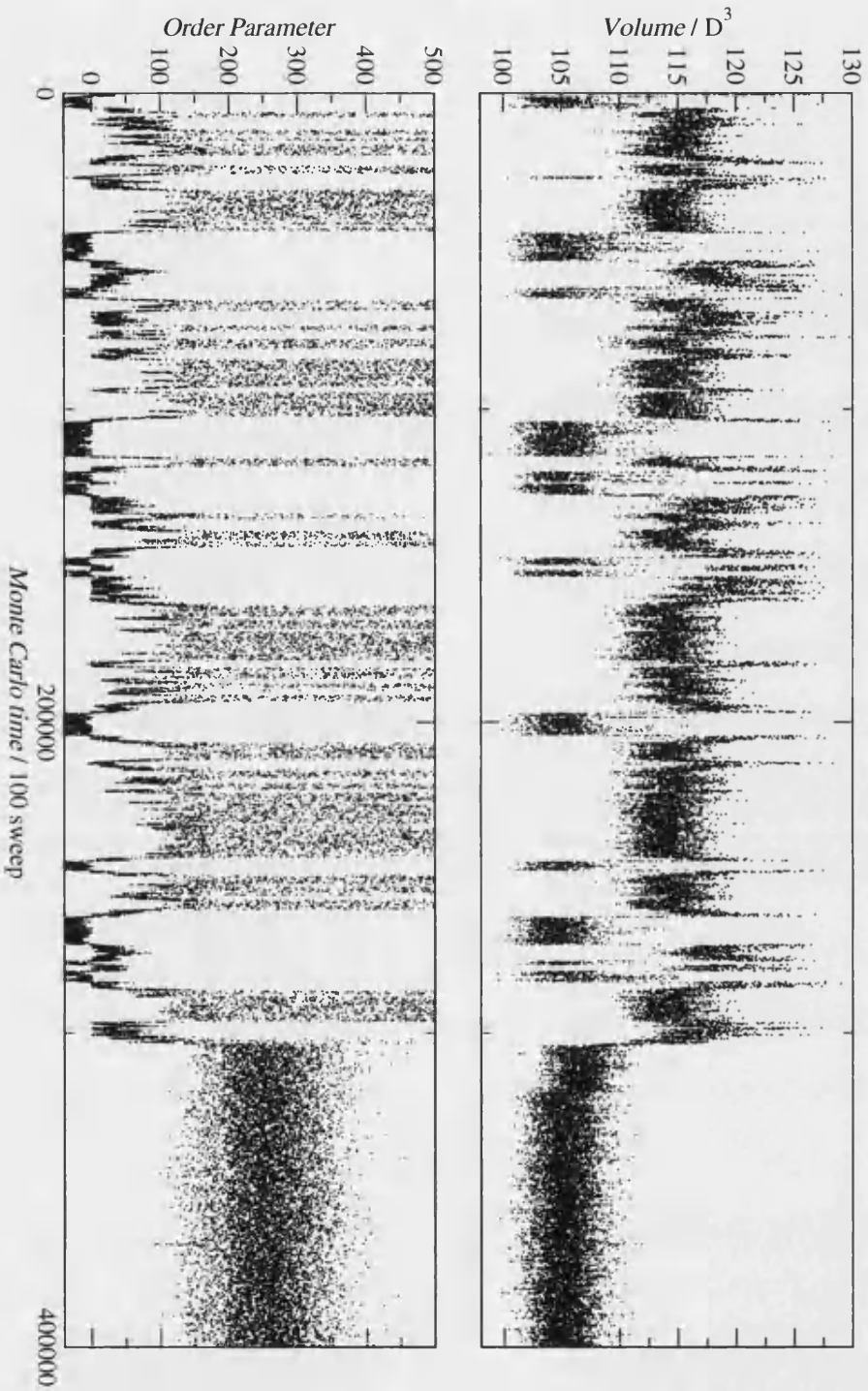


Figure 5.7: Typical Monte Carlo time evolution of system showing the order parameter (shown negative for the solid phase) and volume.

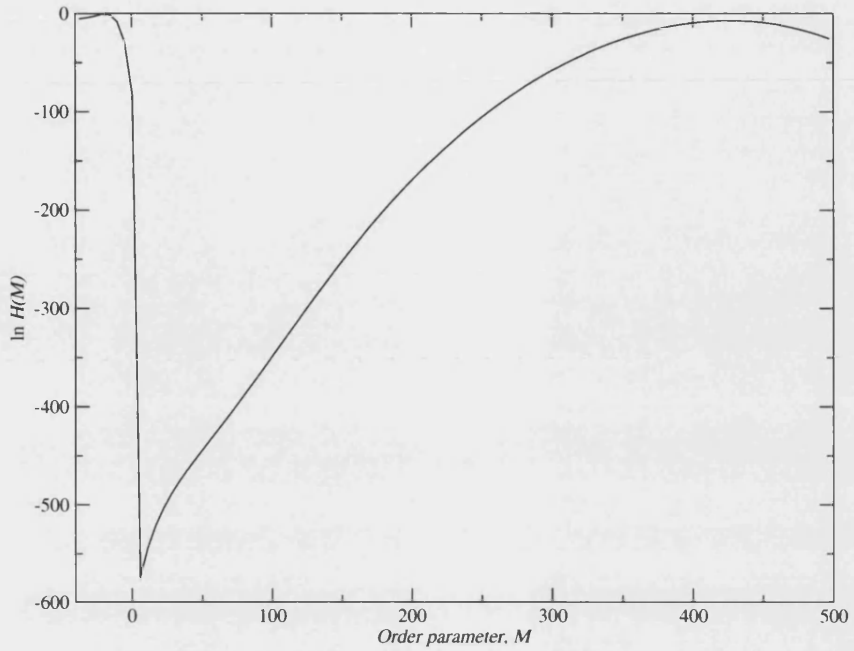


Figure 5.8: Logarithm of the unfolded order parameter histogram ($\ln H(M)$), with no reweighting applied. The fluid phase peak would not be visible on a plot of $H(M)$.

the data sets and excluding data where this had happened.

From the data a coexistence pressure estimate of $P^*_{\text{coex}} = 10.97(8)\ D^{-3}$ was determined. This appears consistent with [106]. In addition, the order parameter and density distributions were compared and again demonstrated good agreement. This suggests that the program is operating correctly. The unfolded order parameter distribution at the original simulation pressure ($P^* = 11.5\ D^{-3}$) is shown in figure 5.8. The same data reweighted to the coexistence pressure estimate above is shown in figure 5.10. The corresponding volume distribution is shown in figure 5.9. The coexistence pressure was also determined by Errington [73] via phase switch, yielding a value of $P^*_{\text{coex}} = 11.00(6)$ again consistent with these results.

5.9 Conclusion

The results show good agreement with the previous work. The problem with spontaneous freezing is not anticipated in large systems, and is manageable in small systems, so no more specialised treatment is necessitated. Simulating closer to the actual coex-

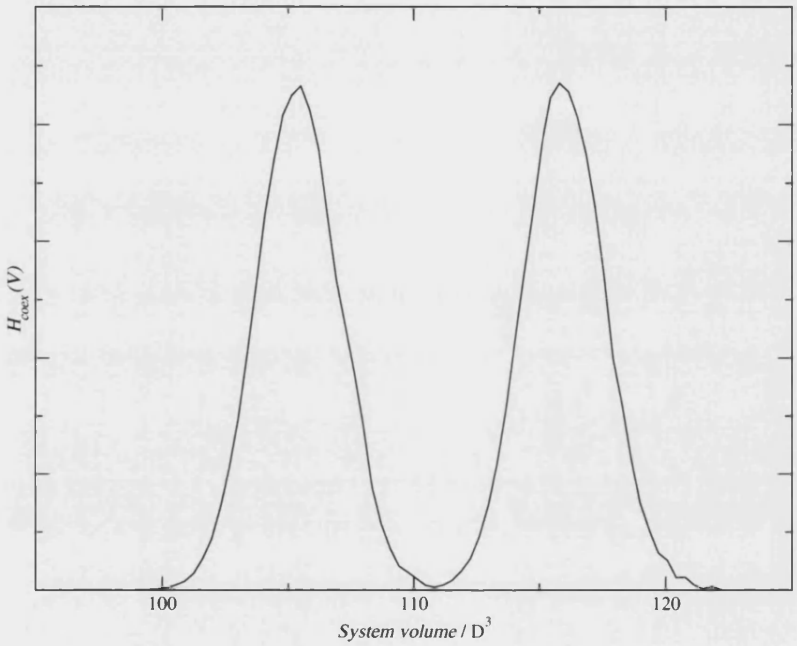


Figure 5.9: Unfolded histogram of the volume distribution, reweighted to coexistence. The left hand peak is the FCC crystalline solid, and the right hand one is the fluid phase.

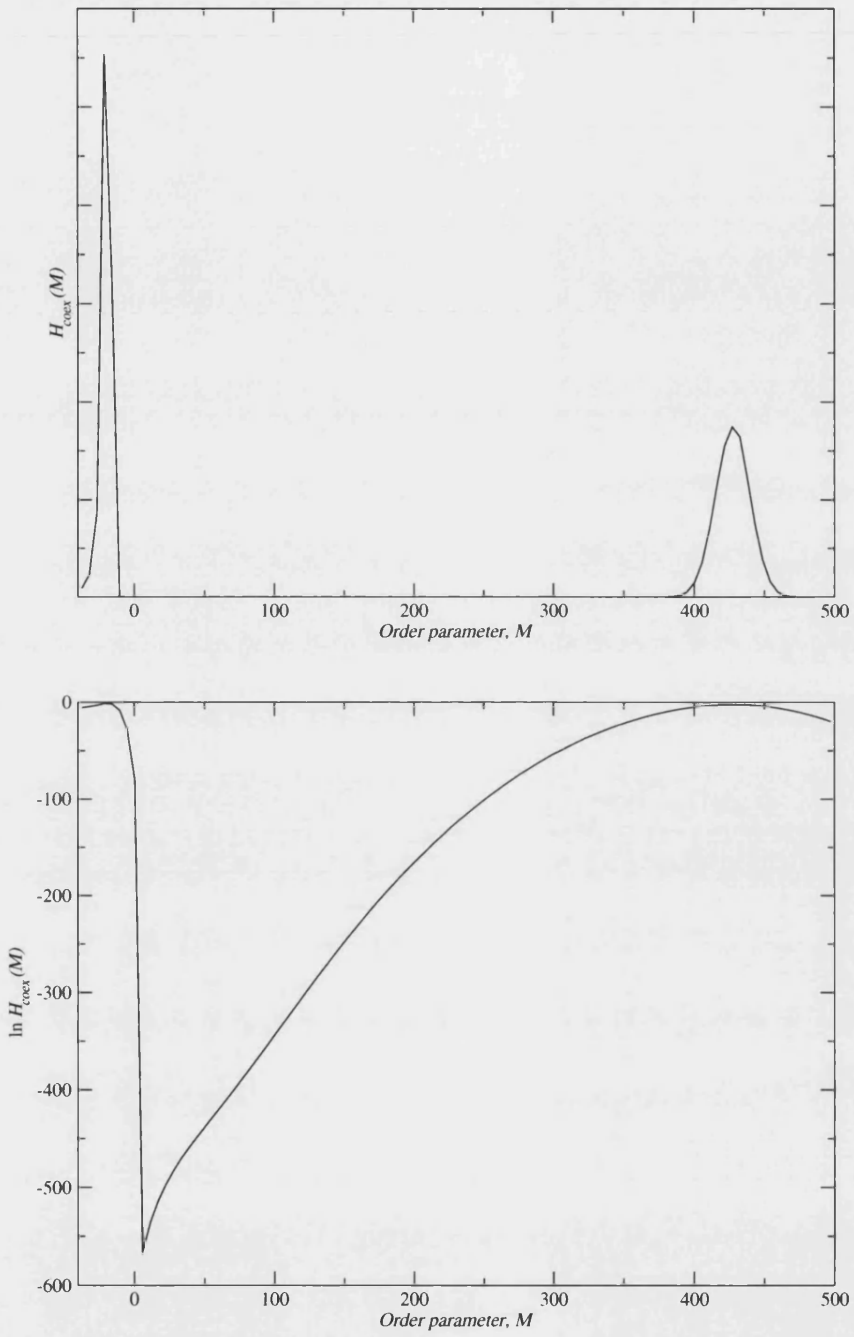


Figure 5.10: Unfolded histogram of the order parameter distribution, reweighted to coexistence. Both the histogram and its logarithm are plotted. The area under the two peaks is (necessarily) equal.

istence pressure would also have alleviated the problem somewhat. This study has laid the groundwork for extension to other, more complicated, systems. The focus of this chapter was developing a new code framework, validated by previous literature results, and also to explore the phase-switch method generally, with the eventual aim of a more automated process.

Much of the computational workload was in generating the weight functions. The performance of the Wang-Landau algorithm on this (much more difficult) system was somewhat disappointing, but proved sufficient for the small system studied.

Chapter 6

The Lennard-Jones freezing line

This chapter extends the work of the previous one by generalising the method to soft potentials. The methodology was developed independently and concurrently with another phase switch Monte Carlo study of the same system [73], but was a more extended study, aiming to determine a large portion of the phase boundary (for accessible system sizes).

6.1 Review

The first major attempt to determine the Lennard-Jones freezing line was the comprehensive series of studies by Kofke [41, 42] and later Agrawal and Kofke [43], in which thermodynamic integration (see §2.2.1) was used to locate an initial point on the freezing line, and Gibbs-Duhem integration (see §2.2.2) was used to map the phase boundary itself. Lennard-Jones was treated as a perturbation to an inverse power law potential rather than explicitly simulated. Honeycutt and Andersen undertook a study of the freezing of small isolated clusters of Lennard-Jones atoms [108] though, as in the hard sphere case, this does not provide a route to the bulk properties. Morris and Song studied Lennard-Jones with a large molecular dynamics simulation including explicit interfaces between the two phases [109]. Finally, both Errington [73] and the study described in this chapter [110] have applied phase switch Monte Carlo to the problem. A previously lattice switch study [103] has shown that the FCC is the stable phase over the vast majority of the phase diagram so we will examine only the FCC–fluid transition.

6.2 Introduction

In this study a comprehensive mapping of the Lennard-Jones phase diagram were attempted via phase switch Monte Carlo. A new order parameter and weight function was required. Rather than commit to a costly weight function determination at every statepoint examined on the freezing curve, instead multiple histogram reweighting was employed to reweight the weight function itself to the next point along the curve, yielding a tracing procedure similar in operation to Gibbs-Duhem integration, but maintaining the thermodynamic link between the phases throughout. We feel one of the great strengths of the technique is the transparency of the error analysis. Since the phases have been linked in a single simulation we can apply well understood and theoretically sound analytical techniques such as block averaging to yield error bounds. Further, the technique can potentially compute the coexistence properties to arbitrary precision; if a more accurate result is required one has only to run the simulation for longer.

This undertaking proved computationally taxing and a number of optimisations were required in order to produce results within a reasonable time-frame. They are discussed in later sections.

6.3 Statistical mechanics

In this section we review the statistical mechanics for this system. The 12-6 Lennard-Jones potential:

$$U(r) = 4\epsilon \left[\left(\frac{1}{r}\right)^{12} - \left(\frac{1}{r}\right)^{6} \right] \tag{1}$$

was simulated in the isobaric–isothermal (NPT) ensemble using a cubic cell with periodic boundary conditions. The potential was truncated at half the box length and a mean field tail correction [107] of:

$$\mathcal{U}_{\text{tail}}(r_c) = \frac{N}{2} \int_{r_c}^{\infty} U(r)\, \rho\, 4\pi r^2 \, \mathrm{d}r = 8\pi N \rho \left(\frac{1}{9\, r_c^9} - \frac{1}{3\, r_c^3} \right), \tag{2}$$

was applied, with r_c the instantaneous cutoff radius. All energy values in this chapter are reduced by ϵ so they can be expressed in a dimensionless form. The tail correction was recalculated after each volume move. The truncation and tail correction were chosen to facilitate comparison with the Agrawal and Kofke study [43]. The total

internal energy, $\mathcal{U}$, becomes:

$$\mathcal{U}(\mathbf{r}^N; L) = \mathcal{U}_{\text{tail}}(L/2) + \sum_{<ij>} U(r_{ij})\,\theta(r_c - r_{ij}), \tag{3}$$

where r_{ij} is the minimum image convention separation of particles i and j modulo the periodic boundary conditions, and $\theta(\cdot)$ is the Heaviside step function. In turn, the sampling distribution is:

$$\mathcal{P}(\Gamma) = \frac{1}{Z} V(\Gamma)\, e^{-\beta\mathcal{H}(\Gamma)-\eta(\Gamma)}, \tag{4}$$

where the factor of V (the volume) is due to fixing one of the particles to prevent the entire lattice drifted relative to the underlying reference state (see §5.5). In this ensemble the Hamiltonian is given by $\mathcal{H} = \mathcal{U} + PV$, where P is the applied pressure and $\mathcal{U}$ is the configurational energy. The partition function follows as:

$$Z = \int_0^\infty \mathrm{d}V\, V e^{\beta PV} \prod_{i=2}^{N} \left\{ \int\int\int_V \mathrm{d}\mathbf{r}_i \right\} e^{-\beta\mathcal{U}(\mathbf{r}^N;V)-\eta_{\gamma,\mathcal{R}}(\mathbf{r}^N;V)}. \tag{5}$$

6.4 Scaled coordinates

The configurations of both phases were tracked during the simulation, proceeding in the usual manner in the active phase, while the conjugate phase was maintained purely in order to allow evaluation of the order parameter. The fluid phase reference state, $\mathbf{R}^{\text{F}}$ used was just a randomly selected microstate, while the solid phase one, $\mathbf{R}^{\text{CS}}$ was a perfect (zero temperature) crystal lattice sized to match the mean system volume of an equilibrium simulation in that phase.

Unlike in the hard sphere case, the u-vectors were scaled by the box length ratio in the conjugate phase:

$$\begin{aligned} \mathbf{r}_i^{\gamma} &= \mathbf{R}_i^{\gamma} + \mathbf{u}_i \\ \mathbf{r}_i^{\gamma'} &= \mathbf{R}_i^{\gamma'} + \frac{L_{\gamma'}}{L_{\gamma}}\mathbf{u}_i. \end{aligned} \tag{6}$$

This modification impacts the phase switch move as well as the calculation of the order parameter. A given microstate a certain 'distance' from the reference state in the fluid phase will map to one that is 'nearer' to the corresponding reference state in the solid phase. Since the fluid has a more open structure and the solid a closer one, this behaviour seems reasonable. It also allows some simplification and has efficiency benefits, which are expounded in the following sections.

It proved expedient to switch the internal representation of the particle coordinates, reference state and u-vectors from being in units of σ to units of the box length itself. With this choice, coordinates within the simulation cell are stored as floating point numbers in the interval [0, 1] rather than [0, L], and the scaling relation of equation 6 becomes:

$$\begin{aligned} \tilde{\mathbf{r}}_i^{\mathrm{F}} &= \tilde{\mathbf{R}}_i^{\mathrm{F}} + \tilde{\mathbf{u}}_i \\ \tilde{\mathbf{r}}_i^{\mathrm{CS}} &= \tilde{\mathbf{R}}_i^{\mathrm{CS}} + \tilde{\mathbf{u}}_i \end{aligned} \tag{7}$$

where a tilde indicates a quantity expressed in units of the box length (i.e. $\tilde{x} = x/L_\gamma$). This approach has some computational benefits:

- Position vectors can be made periodic simply by rounding to an integer and subtracting.
- Dilations (volume moves) can be performed without having to change any of the stored coordinates.
- The length scaling mentioned in the previous section becomes almost automatic to implement.

The translation moves themselves are still performed in real space coordinates, that is the cube over which trial translations are selected remains unscaled.

6.5 Exploiting power law potentials

If instead of calculating and tracking the total energy at each stage of the simulation, we choose instead to accumulate the r^{-12} and r^{-6} terms separately, expressed in units of the box length:

$$\mathcal{U}_{12} \equiv \sum_{<ij>} \left(\frac{L}{r_{ij}}\right)^{12} \quad \text{and} \quad \mathcal{U}_6 \equiv \sum_{<ij>} \left(\frac{L}{r_{ij}}\right)^{6}, \tag{8}$$

then the actual energy can be recovered using:

$$\mathcal{U} = 4\epsilon \left[\frac{\mathcal{U}_{12}}{L^{12}} - \frac{\mathcal{U}_6}{L^6}\right]. \tag{9}$$

The quantities $\mathcal{U}_{12}$ and $\mathcal{U}_6$ are invariant under a volume dilation, permitting an important optimisation. For Lennard-Jones (and other potentials expressible as a linear sum

of power laws) a volume move can now be attempted merely by re-evaluating equation 9 and applying the Metropolis rule. A full $\mathcal{O}(N^2)$ energy calculation is replaced with an $\mathcal{O}(1)$ calculation. In practice this is a key refinement, netting an approximate 40% reduction in runtime (dependent on the frequency with which volume moves are attempted).

The inclusion of a cutoff has the potential to invalidate the above, but by adopting a cutoff proportionate to the box length rather than a fixed radius, we have ensured that volume moves cannot move a particle inside or outside of the cutoff radius.

6.6 Constructing an order parameter

The order parameter definition of §5.4 requires revision before it can be applied to soft-core systems. Two entirely separate **regimes** are introduced, roughly equivalent to the tether and overlap modes for each particle in the hard sphere case (see §5.4), but applied unilaterally to all particles. If a particle lies within a certain distance of its reference site then it is said to be within its **tether radius**. Now when *every* particle lies within its corresponding tether radius, then the system is in **energy mode**. In energy mode we adopt a new order parameter derived from the energy that the system would have *after* a phase switch had been performed. This is similar to the operation of the overlap part of the hard sphere order parameter. In the tether mode, a bias based linearly on the distance from the reference site is applied to each particle, as before. This arrangement is explained in detail below. The same separation was made in Errington's study [73].

6.6.1 Tether order parameter

The **tether order parameter**, T, was implemented as a sum of the distances of each particle from its corresponding reference site (approximately a phase space Hamming distance), but with the contribution from inside the tether radius, r_T, subtracted off. With this definition the order parameter will become zero if and only if all the particles are inside their tether radii. It also proved useful to include a small bias, T_{bias}, acting against each particle outside its tether radius, as this encouraged particles to stay inside their tether radii rather than fluctuate around the perimeter and reduce the probability of a switch to energy mode. The contribution to T from each individual particle, t_i, is shown as a function of the distance from the associated lattice site in figure 6.1.

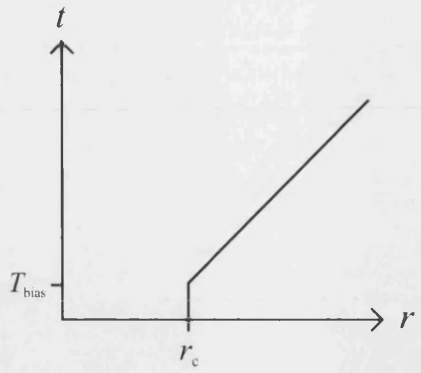


Figure 6.1: The contribution to the T order parameter due to each particle, as a function of the distance from its associated lattice site.

The order parameter was defined to be volume independent, with both the u-vectors and tether radius being expressed in units of the box length, again licensing some simplification (see §6.4). It also serves to suppress the tendency of the weight function to compress the box as a whole, to achieve an overall reduction of T, which amounts to an additional artificial pressure that could impede the kinetics. The full definition of the tether order parameter emerges as:

$$\mathcal{T} = \frac{1}{N}\sum_{i=1}^{N} t_i \quad \text{where:} \quad t_i = \begin{cases} 0 & \text{iff } |\tilde{\mathbf{u}}_i| < r_T \\ |\tilde{\mathbf{u}}_i| - r_T + T_{\text{bias}} & \text{otherwise} \end{cases} \tag{10}$$

The $1/N$ term was included so the order parameter took on similar values in equilibrium simulations of all system sizes. This definition proved adequate to enable the simulation to reach the energy mode, albeit only after the bias term was introduced.

6.6.2 Energy order parameter

For the energy mode, it was necessary to define an *energy order parameter* designed to bias the simulation to reach a state which corresponded to low energy in the conjugate phase as well as for the current phase. In these conditions a phase switch can be successfully launched. The definition used was an extension of the one used by Jackson *et al* [103] in the context of lattice switch. First, the energy of each of the reference states is measured. Then an *excess energy*, $\mathcal{E}_\gamma$, is defined as the energy over and above that of the reference state for each phase. The order parameter is then the difference in these excesses:

$$M = \mathcal{E}_\gamma - \mathcal{E}_{\bar{\gamma}} = (\mathcal{H}_\gamma - \mathcal{H}_\gamma^{\text{ref}}) - (\mathcal{H}_{\bar{\gamma}} - \mathcal{H}_{\bar{\gamma}}^{\text{ref}}).$$

Note that the pressure and volume enter via the Hamiltonian. In the fluid (solid) phase M will in general be very large because the solid (fluid) phase configuration will contain high energy 'overlaps'. Only when the simulation is very near the reference state will M become small. States in the vicinity of $M = 0$ will have a reasonable acceptance in both directions.

6.6.3 Order parameter regime transitions

There are potentially two ways of enabling the simulation to switch from the tether to energy regime and vice versa. Consider the case when all the particles except one are within their tether radii. Then suppose a translation (or other Monte Carlo move) was proposed that happened to bring the last particle inside its tether radius. The two ways of handling it are:

1. **Implicit switch**: the switch is made implicitly as part of the move. The weight for the proposed state is calculated in the energy regime whereas the weight for the current state is calculated in the tether regime. The move is accepted or rejected in the usual way. If it is accepted, then the simulation is now in the energy regime.

2. **Explicit switch**: the suggested move does not automatically change the regime, rather the simulation continues in the tether regime, despite the fact that all particles are now inside their tether radii. To change the regime, another Monte Carlo move, a **mode switch**, is added. This periodically tries to switch to the other mode. It is, however, only ever proposed from states where all the particles are within their tether radii. This move is accepted simply with the ratio of the multicanonical weights of the two states.

Both schemes were implemented but the implicit switch, although requiring a more convoluted implementation, was found to be superior because:

- it was easier to validate correctness,
- it was more efficient,
- and it maintains a desirable one-to-one mapping from microstates to order parameter values.

6.6.4 Transforming the order parameters

Each different phase or regime will require a separate weight function. It is convenient to be able to store the weight function internally by dividing the useful range into a linear array of bins, with a uniform weight applied to each. Interpolation between the bins was avoided. The implementation of the weight function algorithms is then straight forward.

A problem arose when M and T were directly equated to bins: the weight function became unacceptably steep as the $T = 0$ state was approached. The M order parameter varied over many orders of magnitude as deep conjugate phase overlaps boosted the value. Under these circumstances the use of a simple linear 'staircase' weight function was not sufficient to resolve the true weights without using an unfeasibly large number of bins. Rather than change the linear distribution of bins (which would be a significant complication), the order parameters themselves were rescaled. The following (monotonically increasing) transformation function was chosen for the tether order parameter:

$$\mathcal{T} = \ln(T + 1). \tag{11}$$

Note that this still maps $T = 0$ to $\mathcal{T} = 0$. The energy order parameter was also similarly rescaled:

$$\mathcal{M} = \text{sign}(M) \ln(|M| + 1), \tag{12}$$

again a linear grid in $\mathcal{M}$ better resolves small M. The use of linear interpolation could reduce the required number of bins further.

Order parameter values lying outside the range explicitly tailored for by the bin arrays must also be handled gracefully. Any $\mathcal{T}$ greater than a maximum value (established from a preliminary equilibrium simulation) is given the same weight as the last bin. However any of the schemes that update weights interactively (such as Wang-Landau) do not change the weight while the simulation is in this region (hence this system is distinct from just making the last bin very wide). The same mechanism was repeated for both ends of both energy order parameters, except that the lower limit was fixed at zero.

6.6.5 Using tethers in crystalline solids

There are two possible ways to choose the tether radius, $\tilde{r}_T$, and corresponding layout of the order parameters.

- **Three mode formulation**: $\tilde{r}_T$ is set similar to the inter-particle spacing and the simulation is constrained to sample only energy mode in the solid phase, effectively tying each particle to its associated lattice site. The tether radius can be made large enough that this does not affect simulation results. A wide range of $\mathcal{M}$ values will be sampled.
- **Four mode formulation**: $\tilde{r}_T$ is set much smaller, necessitating the use of tether mode in the solid phase, but restricting $\mathcal{M}$ to a narrower range of values.

The choice between them will depend on the radius at which energy mode becomes more efficient than tether mode. Computer experiments have shown that in general the four mode formulation is superior because the energy mode is not efficient when high energy values are involved. The trade-off is between the tether order parameter which is effective at locating the gateway states (penetrating the entropic barrier) versus the energy mode which excels at exploring the vicinity of the gateway states efficiently, and providing a bias based on the actual phase switch acceptance used thus directly facilitating the phase switch itself. Errington chose the three mode formulation but simultaneously softened the core of the potential for the conjugate phase only (contrast with the lattice switch order parameter of §5.3).

For the smallest system size it proved not only possible, but highly desirable to completely eliminate the energy mode by reducing the tether radius even more and attempting phase-switches directly in tether mode.

6.7 Mutual energy table

Another optimisation, which was less valuable than some of the others was the introduction of a **mutual energy table** containing the interaction energy for every pair of particles in the system (there are $N(N-1)/2$ such terms). The advantage of this is that, for each translation move attempted, the terms contributing to the energy of a particle can be easily identified and summed, saving both the calculation of the separation modulo the periodic boundary conditions and the evaluation of the potential itself. This comes at the cost of a memory lookup and extra book-keeping (all interaction terms involving a particle must be updated each time a move is accepted).

To avoid updating the entire table after every volume move, the $\mathcal{U}_{12}$ and $\mathcal{U}_6$ terms (see section 6.5) for each interaction are both stored in the table, rather than storing the actual energy. The table also contains these terms for the conjugate phase, although those are only kept updated in the energy regime.

It was found that the mutual energy table conferred an advantage of 30% of run-time for the $N = 108$ system, was of negligible advantage in the $N = 256$ system and was actually slower for larger systems. This derives from a combination of the memory access becoming increasingly expensive as the table grows (due to more cache memory misses) but also to the time needed to update the table after a move is accepted.

6.8 Integer representation of particle coordinates

Typically about half the simulation time is spent enforcing the periodic boundary conditions, so it seems amenable to optimisation. If we store particle coordinates as unsigned integer values such that the cell width corresponds to the full representable range of the integer (2^{32} typically) then we can exploit the overflow and underflow behaviour of current computers to effectively get the periodic boundary calculation for free. The values are converted back to floating point for the energy calculation itself. Nevertheless this was an important performance gain, roughly doubling the speed of the simulation.

6.9 Radial moves

One improvement that was considered was adding a new type of translation move, used in addition to the usual uniform cubic moves. These were translations along the tether connecting a particle to its reference site, either moving it further from, or closer to, its associated lattice site along the radius. The motivation was to preferentially attempt moves that attacked the $\mathcal{T}$ order parameter directly. The cubic moves have difficulty reaching the gateway states because it is typically only a small fraction of the proposal cube that would reduce the tether order parameter. By contrast these radial moves act directly on the order parameter directly, with half the proposed moves being to states of smaller $\mathcal{T}$. It was hoped that this would speed up the slowest (longest correlation time) section of the order parameter space.

6.9.1 Derivation

We consider radial moves that are applied as a scaling around the reference site, mapping $\mathbf{r} = \mathbf{R} + \mathbf{u}$ onto $\mathbf{r}' = \mathbf{r} + s\mathbf{u}$; where we call s the scaling factor. The scaling factor is chosen randomly for each move up to some maximum, $s_{\max}$, where we assume

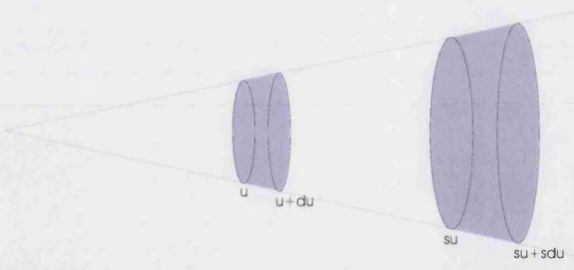


Figure 6.2: A proposed radial move from differential region around $\mathbf{r}$ to differential region around $\mathbf{r}'$.

$1 < s_{\max} < 2$.

We chose s thus: a uniform random deviate δ is generated in the interval $[-\ln s_{\max}, \ln s_{\max}]$. Then s follows as e^{δ}. A move with random number δ is exactly opposed to one with $-\delta$ and so they are balanced. This is equivalent to pairing a move with scale factor s is with one with scale factor $1/s$.

Now consider a move from a microstate Γ with u-vector for particle i of $\mathbf{u}_i$ using a scale factor of s to a create a new microstate Γ' which is identical except that $\mathbf{u}'_i = s\mathbf{u}_i$ (where primes indicate quantities after the move). Consider the phase volume surrounding each of these microstates: to ensure symmetry we take the geometry of figure 6.2. Here the differential element is a section of the cone of solid angle ϕ around $\mathbf{u}_i$, at $\mathbf{u}_i$ with a width of $\mathrm{d}u$ (see above). Approximating volume as surface area multiplied by depth we get a phase space volume of $\phi u^2 \mathrm{d}u$, which becomes exact as $\mathrm{d}u \to 0$.

Under the move that volume projects onto $\phi(su)^2(s\,du) = \phi s^3 u^2 du$. Detailed balance (equation 15 on page 23) becomes:

$$\pi(\Gamma \to \Gamma')a(\Gamma \to \Gamma')\mathcal{P}(\Gamma)\mathrm{d}\Gamma = \pi(\Gamma' \to \Gamma)a(\Gamma' \to \Gamma)\mathcal{P}(\Gamma')\mathrm{d}\Gamma' \tag{13}$$

The π terms cancel (only with $s = e^{\delta}$ however). Rearranging and substituting $\mathrm{d}\Gamma/\mathrm{d}\Gamma' = 1/s^3$, and substituting $\mathcal{P}(\Gamma) = \eta(\Gamma)V\,e^{-\beta\mathcal{H}(\Gamma)}$ gives:-

$$\frac{a(\Gamma \to \Gamma')}{a(\Gamma' \to \Gamma)} = \frac{V' e^{-\beta\mathcal{H}(\Gamma')}}{V e^{-\beta\mathcal{H}(\Gamma)}}\, s^3\, e^{\eta(\Upsilon)-\eta(\Gamma)}$$

Noting that $V = V'$ and $\Delta\mathcal{H} = \Delta\mathcal{U}$, a Metropolis acceptance of:

$$a(\Gamma \to \Gamma') = \min\left\{1,\, s^3 e^{-\beta[\mathcal{U}(\Gamma')-\mathcal{U}(\Gamma)]}\, e^{\eta(\Upsilon)-\eta(\Gamma)},\right\} \tag{14}$$

is obtained.

The s^3 term can be understood as the product of two effects. Firstly, that we are moving between shells around the reference site and thus the shell surface area scales as s^2. To create a balanced move we must increase the chance of moving to the larger phase volume by that factor. Secondly, by doing a scaling rather than translation we are mapping a small differential, du, along the u-vector onto a larger one further out, yielding the third factor of s.

Empirically it was found that the radial moves offered only marginal gains over cubic translation moves, and they were abandoned as an unnecessary complication. We speculate that any moves which are attempting to overcome an entropic (as oppose to kinetic) barrier are in a sense doomed to fail since we are bound by the detailed balance condition to force a low acceptance in proportion to the compression of phase space afforded by the move (note the appearance of the s^3 term in equation 14). This does not invalidate the efficacy of novel moves which span large regions of phase space, such as the generalised geometric cluster algorithm [27], which is still adiabatic with respect to phase space. Novel moves are suggested when attacking problems of slow kinetics.

6.10 Simulation details

The simulation was performed at a total of four different system sizes: $N = 32, 108, 256$ and 500 corresponding to 2, 3, 4 and 5 unit cells (of four atoms) per axis. The $N = 32$ system was small enough that real freezing and melting events occurred with sufficient regularity that the phase switch formalism was unnecessary. Instead we assigned an instantaneous phase label to the simulation based purely on the density. Since the density distributions of the phases overlapped significantly, it was necessary to form a block average over 50 consecutive measurements (5000 cycles) to correctly discriminate the phases.

The simulation itself consisted of a cycle of $N - 1$ translation attempts, N association swaps, one volume move and one phase switch. When radial moves were included, a

sweep of $N-1$ trials was added. The acceptance rules used were:

$$\begin{aligned}
&\text{Particle translations} && a(\Gamma \to \Gamma') = \min\left\{1,\ e^{\eta_{\gamma',\mathcal{R}'}(M_{\gamma',\mathcal{R}'})-\eta_{\gamma,\mathcal{R}}(M_{\gamma,\mathcal{R}})}\ e^{\beta(\mathcal{U}'-\mathcal{U})}\right\}\\
&\text{Volume moves} && a(\Gamma \to \Gamma') = \min\left\{1,\ e^{\eta_{\gamma',\mathcal{R}'}(M_{\gamma',\mathcal{R}'})-\eta_{\gamma,\mathcal{R}}(M_{\gamma,\mathcal{R}})}\ e^{\beta(\mathcal{U}'-\mathcal{U})}\right\}\\
&\text{Association swaps} && a(\Gamma \to \Gamma') = \min\left\{1,\ e^{\eta_{\gamma',\mathcal{R}'}(M_{\gamma',\mathcal{R}'})-\eta_{\gamma,\mathcal{R}}(M_{\gamma,\mathcal{R}})}\right\}\\
&\text{Phase–switch} && a(\Gamma \to \Gamma') = \min\left\{1,\ e^{\eta_{\gamma',\mathcal{R}'}(M_{\gamma',\mathcal{R}'})-\eta_{\gamma,\mathcal{R}}(M_{\gamma,\mathcal{R}})}\left[\frac{V'}{V}\right]^{N+1} e^{-\beta[\mathcal{H}-\mathcal{H}']}\right\}.
\end{aligned} \tag{15}$$

The phase switch acceptance is derived in appendix D and differs from that of the previous chapter because of the inclusion of a dilation in its formulation in this study. The various branches of the order parameter were organised as is laid out in §6.6.4.

Various observables were sampled throughout the simulation, with sampling every 25 or 100 cycles. This data included the current order parameter value and mode as well as the volume and energy so that full histogram reweighting could be performed, including of the weight function itself. In all, around 80 GB of data was collected for this study.

The reference lattices used were optimised: the FCC reference lattice was taken as a perfect crystal, but scaled to have the equilibrium density. The fluid phase reference state was derived by quenching a liquid state reference lattice down to zero temperature via a simulated annealing process, with an extra constraint to prevent atoms moving more than a short distance from their start positions. These steps were shown to be of modest benefit to the overall simulation.

6.11 Determining the weight functions

The biggest challenge was initiating the phase switch runs, which required construction of a large, deep weight function capable of biasing a simulation to visit the gateway states. Following the difficulties encountered in chapter 5 using the Wang-Landau algorithm the transition matrix method was adopted instead (see §3.2.3) with an iteration interval of 20000-50000 cycles. This proved sufficient for the task, although early experiments which combined linear interpolation of the weight function with the transition matrix method failed because the derivation of the transition matrix method assumes a uniform weight for the entire macrostate, underlining the need to carefully check the

assumptions of all techniques applied to a particular problem in the context of that problem. Despite the efficiency of the method, windowing was still required, with up to 3 windows (see §3.2.4) used to cover each energy range, and up to 9 required for the fluid phase tether region in the largest system. An example weight function is shown in figure 6.3.

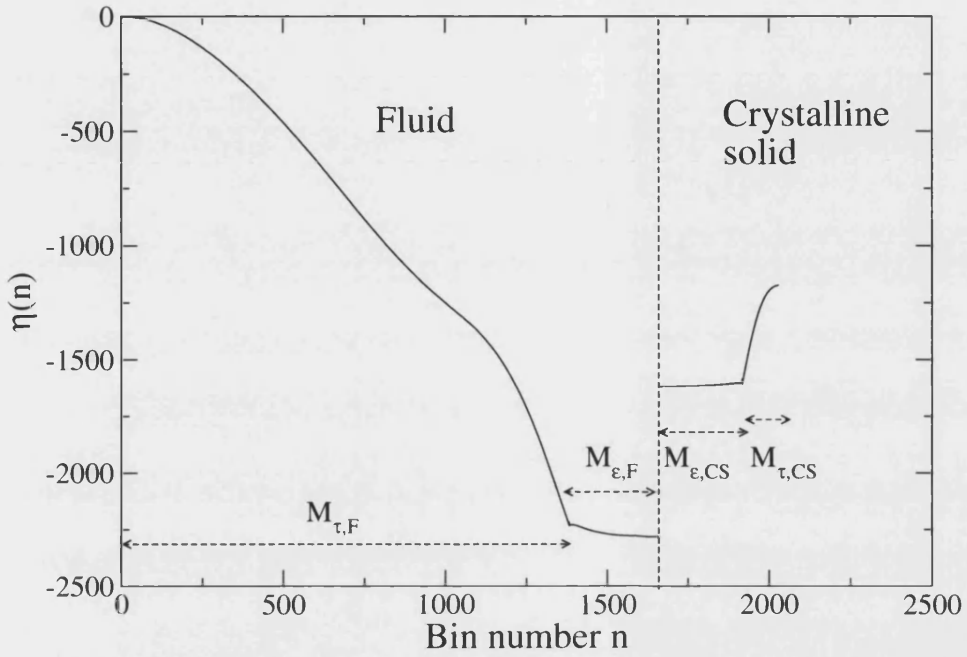


Figure 6.3: An example weight function: the $N = 256$ system with an inverse temperature of $\epsilon/k_BT = 0.6$ and pressure of $P\sigma^3/\epsilon$. The four branches of the weight function are shown with respect to the underlying discretisation into bins.

A further problem is matching the weight function fragments relative to each other. While the tether and energy regions could simply be matched using a window overlapping both, another method was needed to determine the size of the jump required to match the fluid and solid branches together. A 'fuzzy' root finding algorithm was used, which determined the dominant phase by simulating a window encompassing only the section immediately adjacent to $M = 0$ in each phase. The weight function gap was altered to boost the probability of the less dominant phase after each iteration and the boost was reduced by a factor of 50% with each subsequent iteration, eventually converging on the correct value.

6.11.1 Analysis

Multiple histogram reweighting (see §2.1.2) was used to locate coexistence. Consider the function:

$$f(\beta, P) = \frac{p_B(\beta, P)}{p_A(\beta, P) + p_B(\beta, P)} - \frac{1}{2}, \tag{16}$$

with p taken to be the relative statistical weight of a phase with given values of the thermodynamic fields. By fixing either β or P (depending on the exact position in the phase diagram), and varying the other under the control of a root finding algorithm [111] coexistence may be located.

Error bars were calculated by taking subsets of the data and re-computing the coexistence point for each subset and comparing. Initially **block averaging** was attempted, but the exceptionally long correlation length in this system lead to only a very small number of blocks and overly conservative error estimates. Instead, both **jackknife** [112, 113] and **bootstrap** [114, 115] error bounds were computed. It was determined that bootstrap error bounds were the more stable, possessing only a very weak dependence on the block size chosen, and so this procedure was used with block sizes of

6.12 Results and discussion

Most of the simulations were performed on a 96-node parallel cluster computer based on `UltraSPARC-III` technology, but a small cluster of desktop machines was also used, particularly for the $N = 500$ data series. Although the individual cluster nodes were slower than contemporary desktop machines, the ability to launch a series of around 8 runs per statepoint per system size simultaneously afforded a considerable amount of computational power. For the 108 particle system, actual phase transitions occurred with a frequency of around one or two per 24 hour period. For the 256 particle system it was a number of days, and for the $N = 500$ system running on faster hardware it was around a week. Each statepoint was used to extrapolate a new weight function for the next one so the calculation was not further parallelisable without working from multiple starting points on the phase boundary. Determining the 3 or 4 branches of the weight function and stitching them together was more involved, requiring the liquid phase tether region to be split into windows, which could then run in parallel (see previous section). A representative run, plotting order parameter against Monte Carlo time is shown in figure 6.4.

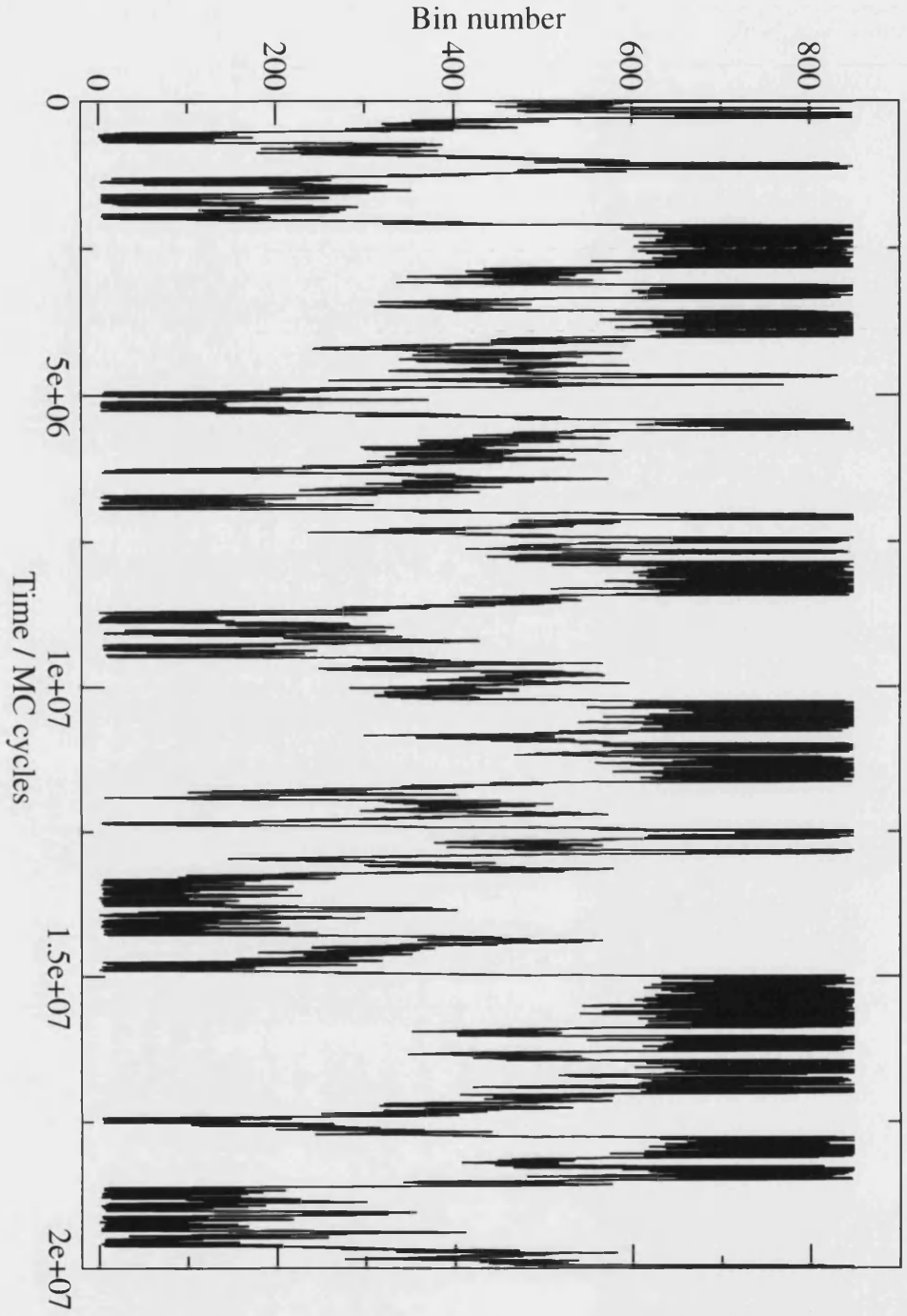


Figure 6.4: Typical example of the evolution of the order parameter (bin number) as a function of Monte Carlo time. This data set is for the $N = 256$ system at an inverse temperature of $\epsilon/k_BT = 1$ and pressure of $P\sigma^3/\epsilon = 3.51686$. The fluid phase appears at low bin number and the solid phase at high bin number.

6.12.1 Spontaneous freezing

One problem encountered was **spontaneous freezing** (previously mentioned in §5.8) where, for small system sizes, the system actually managed to nucleate a solid dynamically. This happened quite often with the $N = 108$ system, very infrequently in the $N = 256$ and never in the $N = 500$ system. The crystal formed was rarely a perfect FCC crystal (evidenced from the density distribution). Naturally, this undermines our strategy of maintaining a current phase label merely by counting the successful phase switches. To deal with this we simply discarded simulation runs where spontaneous freezing had occurred, being careful to also discard a correlation length's worth of data leading up to the spontaneous freezing event so no possible bias could enter the production data. An arbitrary cutoff in some suitable order parameter could have been used to suppress spontaneous freezing, but it was felt the disadvantage of potentially biasing the results was not justified.

The converse, **spontaneous melting** did not occur because we imposed an artificial maximum tether length in the solid phase, necessary in any case to ensure only a single permutation was sampled. Comparison of the u-vector distribution between extended equilibrium and phase switch runs showed that this had no impact on the fluctuations occurring within the crystal.

We did confirm that spontaneous freezing from the fluid phase was reversible, two simulations that were accurately located on the coexistence line were allowed to continue to run and eventually melted back to the fluid state again. As a check against the implementation of the main phase switch method, a direct comparison using one of these reversible runs was made showing agreement within error bars.

In later simulations, particularly those at high temperature, we chose another route to minimise the problem: we lowered the simulation pressure slightly (by about 5%) resulting in a simulation less likely to spontaneously freeze, although the solid phase will still be fully explored by the phase switch method. Such a modification was sufficiently slight to have little impact on the sampled phase space.

6.12.2 Lennard-Jones Phase diagram

A typical density distribution derived from the simulation is shown in figure 6.5. The freezing line for the range of system sizes studied here is presented in figure 6.6, also including both the results of the Kofke and Agrawal study and those of Errington. The phase diagram is shown in the density–temperature plane in figure 6.7. The raw data,

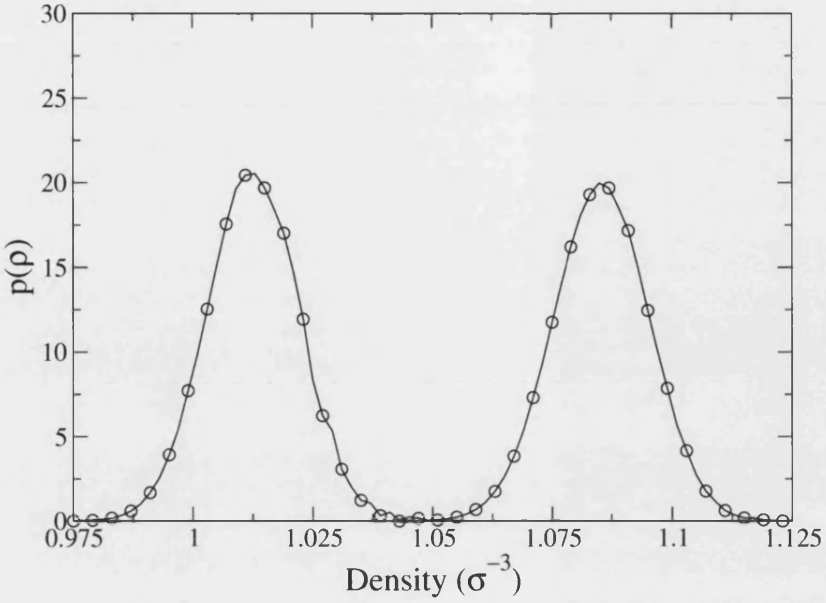


Figure 6.5: The density distribution $p(\rho)$ for a system of $N = 256$ Lennard-Jones particles reweighted to coexistence at $\epsilon/k_BT = 0.6$, $P\sigma^3/\epsilon = 13.722$. A selection of data points are shown. Lines are guides to the eye.

together with uncertainties, is presented in tables 6.1, 6.2 and 6.3.

In the hard sphere case, a finite size scaling relation of $P_{\text{coex}} \propto N^{-1}$ was observed [120, 73], but no such relationship was seen to hold in this study. Presumably this is a result of the scalable truncation radius and mean field tail correction. A further study with the potential defined as in chapter 4 might provide a route to the freezing parameters at the thermodynamic limit.

This study showed good agreement with the results of the Errington study [73], which is to be expected since the same potential truncation and method were used. The results of the Gibbs-Duhem study, by contrast do not agree within error bounds. This may be for a number of reasons: the treatment of Lennard-Jones as a perturbation, drift of the Gibbs-Duhem integration, or an error in the initial state point.

$N = 108$					
ϵ/kT	$P\sigma^3/\epsilon$	$\rho_{CS}\,\sigma^3$	$\rho_F\,\sigma^3$	$\mathcal{E}_{CS}/N\epsilon$	$\mathcal{E}_F/N\epsilon$
0.01250	3489(8)	2.5682(18)	2.5531(10)	1563.4(6)	1572.15(31)
0.01550	2635(9)	2.4141(11)	2.3277(13)	1204.7(3)	1251.84(54)
0.01900	2039(3)	2.3045(4)	2.2197(2)	1008.5(1)	1050.32(8)
0.02350	1549(2)	2.1905(2)	2.1091(1)	813.48(7)	847.951(32)
0.02800	1232(1)	2.1014(3)	2.0212(5)	632.69(7)	660.221(73)
0.03500	922(1)	1.9935(2)	1.9182(3)	503.94(5)	526.283(37)
0.04500	663.1(9)	1.8803(2)	1.8078(3)	389.12(4)	406.995(38)
0.05500	507.7(8)	1.7935(1)	1.7235(2)	312.12(2)	326.930(22)
0.06500	406.5(6)	1.7260(2)	1.6576(1)	257.63(2)	270.276(10)
0.07500	334.6(9)	1.6702(7)	1.6033(3)	219.03(6)	230.064(21)
0.09000	262.0(7)	1.6026(7)	1.5383(5)	169.65(3)	178.562(29)
0.11000	197.7(4)	1.5299(3)	1.4659(2)	135.32(2)	142.881(16)
0.13000	157.8(3)	1.4773(3)	1.4142(2)	110.82(2)	117.339(8)
0.16000	117.2(2)	1.4111(4)	1.3477(6)	82.416(15)	87.742(24)
0.17500	103.0(3)	1.3827(3)	1.3213(5)	72.448(1)	77.294(24)
0.21000	78.95(13)	1.3304(3)	1.2692(1)	56.230(9)	60.335(4)
0.24000	64.68(8)	1.2940(1)	1.2327(1)	47.043(4)	50.738(5)
0.30000	45.66(6)	1.2349(1)	1.1734(4)	30.560(4)	33.483(7)
0.37000	32.43(5)	1.1835(1)	1.1214(1)	22.669(2)	25.210(3)
0.43000	25.12(4)	1.1505(11)	1.0863(4)	16.44(50)	18.718(10)
0.50000	19.13(3)	1.1180(11)	1.0525(3)	11.17(34)	13.163(6)
0.62000	12.56(4)	1.0762(12)	1.0070(3)	5.21(16)	6.908(4)
0.69257	9.840(17)	1.0557(11)	0.9844(2)	2.72(8)	4.271(3)
0.80000	6.859(8)	1.0291(11)	0.9542(2)	-0.23(1)	1.157(3)
0.95000	4.105(10)	1.0023(11)	0.9212(1)	-2.98(91)	-1.735(2)
1.10000	2.236(7)	0.9812(11)	0.8925(2)	-4.85(15)	-3.701(2)
1.20000	1.293(4)	0.9695(11)	0.8741(1)	-5.86(18)	-4.752(1)
1.26282(42)	0.8173	0.9637(11)	0.8645(1)	-6.30(19)	-5.207(1)
1.33510(54)	0.3150	0.9570(11)	0.8519(2)	-6.83(21)	-5.755(1)
1.37193(39)	0.1003	0.9543(11)	0.8464(3)	-7.07(21)	-5.994(2)
1.38077(56)	0.0466	0.9537(11)	0.8447(2)	-7.13(21)	-6.051(2)
1.38564(46)	0.0250	0.9537(11)	0.8447(2)	-7.15(22)	-6.080(1)
1.38674(47)	0.0150	0.9533(11)	0.8438(2)	-7.16(22)	-6.087(2)
1.38963(91)	0.0010	0.9534(11)	0.8439(1)	-7.18(22)	-6.105(1)

Table 6.1: Solid-fluid coexistence curve data for the $N = 108$ system. Tabulated in columns $1-6$, respectively, are the inverse freezing temperature, pressure, solid number density, fluid number density, energy per particle (solid) and energy per particle (fluid). Numbers in parentheses indicate the the 67% confidence limit for the rightmost digit(s). Note that due to the steepness of the coexistence curve at low pressures (cf. fig. 6.6), it is expedient in this regime to determine the coexistence temperature at prescribed pressure.

$N = 256$					
ϵ/kT	$p\sigma^3/\epsilon$	$\rho_{CS}\,\sigma^3$	$\rho_F\,\sigma^3$	$\mathcal{E}_{CS}/N\epsilon$	$\mathcal{E}_F/N\epsilon$
0.39000	30.400(47)	1.1771(3)	1.10965(3)	20.200(22)	22.8182(6)
0.43000	25.390(180)	1.1525(5)	1.08401(12)	15.870(22)	18.2534(30)
0.47000	21.942(37)	1.1355(2)	1.06611(1)	12.430(13)	14.6269(1)
0.54000	16.928(34)	1.1065(2)	1.0355(2)	8.1409(89)	10.0970(21)
0.60000	13.667(49)	1.0854(3)	1.0130(2)	5.4899(69)	7.2971(19)
0.68000	10.470(23)	1.0616(2)	0.9861(5)	3.1561(39)	4.8445(52)
0.68000	10.477(18)	1.0619(4)	0.9864(1)	3.1494(43)	4.8415(15)
0.80000	7.068(12)	1.0332(3)	0.9532(1)	-0.0465(6)	1.4633(7)
0.91000	4.885(13)	1.0126(3)	0.9278(1)	-2.1525(28)	-0.7639(17)
1.00000	3.546(9)	0.9990(3)	0.9098(1)	-3.5496(44)	-2.2385(8)
1.10000	2.340(9)	0.9851(3)	0.8901(1)	-4.7187(56)	-3.4660(11)
1.20000	1.418(6)	0.9745(3)	0.8734(1)	-5.7189(62)	-4.5181(11)
1.29969(94)	0.6456	0.9651(3)	0.8560(3)	-6.5252(70)	-5.3518(26)
1.33481(74)	0.4263	0.9628(3)	0.8508(1)	-6.7631(74)	-5.5961(9)
1.36274(71)	0.2500	0.9604(3)	0.8463(2)	-6.9512(74)	-5.7908(18)
1.38853(64)	0.1000	0.9587(3)	0.8422(2)	-7.1148(76)	-5.9582(11)
1.40025(60)	0.0300	0.9578(3)	0.8405(1)	-7.1896(77)	-6.0372(8)
1.40527(68)	0.0010	0.9574(3)	0.8395(1)	-7.2208(77)	-6.0681(6)

Table 6.2: Coexistence curve for $N = 256$ system. See caption of tab. 6.1 for details.

$N = 500$					
ϵ/kT	$p\sigma^3/\epsilon$	$\rho_{CS}\,\sigma^3$	$\rho_F\,\sigma^3$	$\mathcal{E}_{CS}/N\epsilon$	$\mathcal{E}_F/N\epsilon$
1.336(2)	0.5500	0.9655(1)	0.8566(2)	-6.637(1)	-5.486(2)
1.3613(3)	0.3836	0.96337(4)	0.85274(3)	-6.815(1)	-5.6705(2)
1.393(1)	0.2091	0.9617(1)	0.8490(1)	-7.007(1)	-5.871(1)
1.418(2)	0.0200	0.9581(1)	0.843(1)	-7.198(1)	-6.08(1)
1.3092(2)	0.7166	0.96726(3)	0.86054(3)	-6.4572(7)	-5.3003(3)
1.3362(5)	0.5500	0.96552(5)	0.85695(7)	-6.6381(9)	-5.4876(6)
1.3613(2)	0.3836	0.96337(4)	0.85274(3)	-6.8148(8)	-5.6705(3)
1.3934(6)	0.2091	0.96170(6)	0.84897(6)	-7.0069(9)	-5.8706(5)
1.4121(3)	0.0800	0.95956(3)	0.84472(6)	-7.1407(8)	-6.0067(5)
1.4228(3)	0.0200	0.95881(3)	0.84299(6)	-7.2055(8)	-6.0729(5)

Table 6.3: Coexistence curve for $N = 500$ system. See caption of table 6.1 for details.

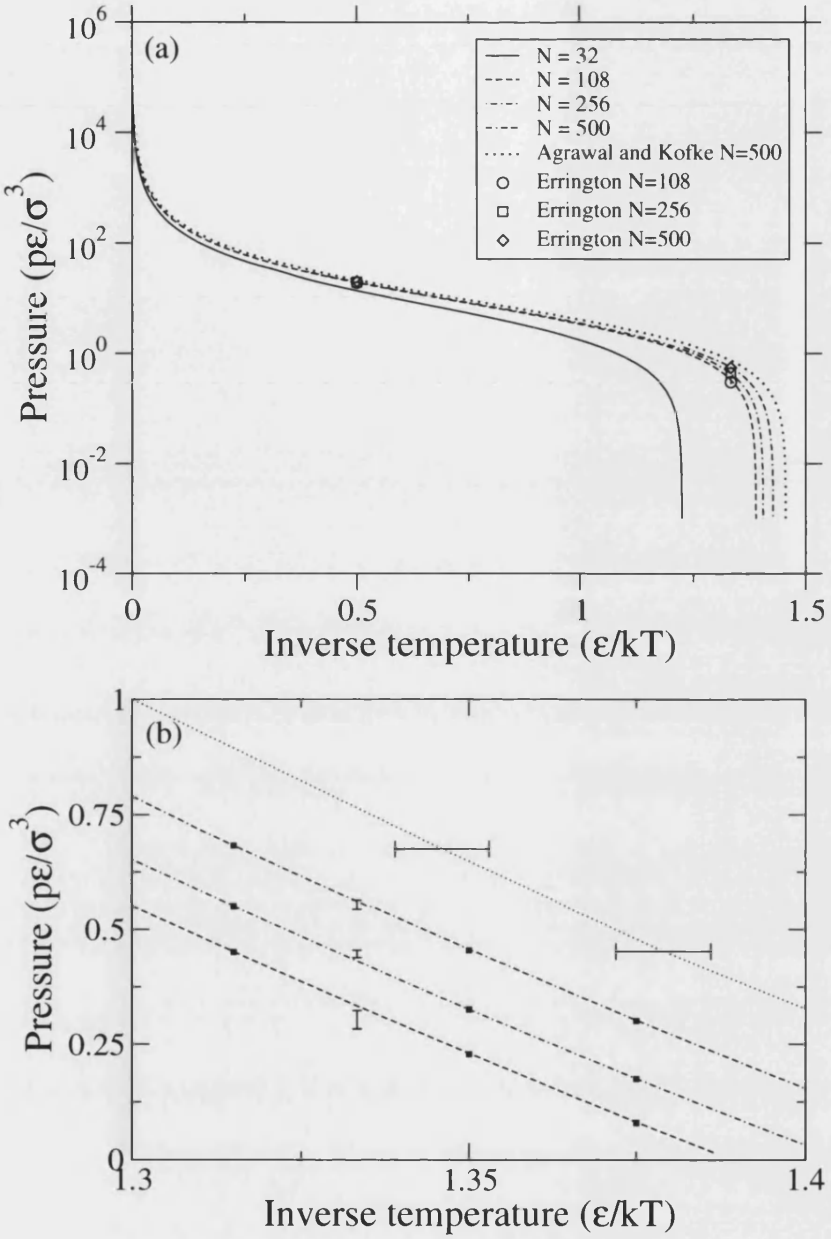


Figure 6.6: **(a)** The Lennard-Jones freezing line in the pressure-inverse temperature plane for the four systems sizes examined. The data shown derive from 20 separate simulation state points for the $N = 32$ system size, 37 points for the $N = 108$ system, 17 points for $N = 256$ and 4 points for the $N = 500$ system size. Also included are the two points of Errington's study [73] and the freezing line from the Agrawal and Kofke [43] study for $N = 500$. **(b)** A closeup of the region around $\beta = 4/3$. The vertical error bars correspond to Errington's data points [73] and the horizontal ones to the Agrawal and Kofke study. Symbols are this study and uncertainties are smaller than the symbol size in each case. Lines are interpolations between the data points, based on multiple histogram reweighting.

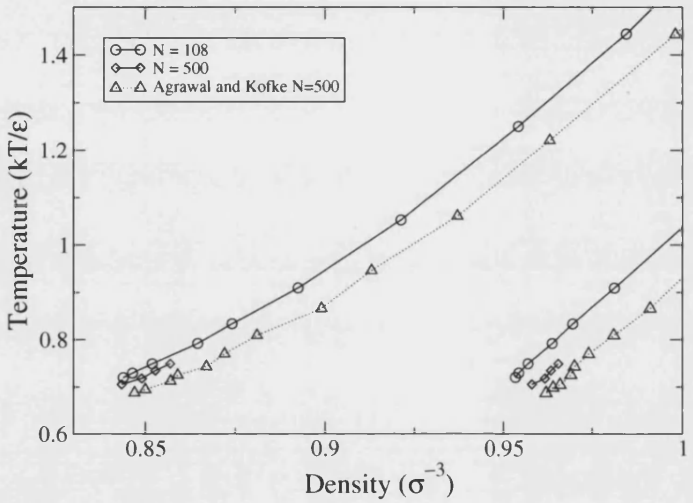


Figure 6.7: A portion of the phase diagram in the $\rho - T$ plane in the region of the triple point. Shown are the estimated solid and fluid coexistence densities for $N = 500$ and $N = 108$. Also included for comparison are the GDI estimates of Agrawal and Kofke [43] for $N = 500$. Uncertainties are smaller than the symbol sizes; lines are guides to the eye

Chapter 7

Conclusion

Extended sampling methods have been used pervasively throughout the studies in this thesis, and the large body of literature on phase equilibria. One of the chief advantages over more indirect methods of obtaining the same quantities is the ability to construct a single simulation which visits all the phase space of interest. This has a number of benefits, but the most striking is the transparency of the error analysis, and the freedom from external approximations (such as integration errors of various kinds). The biggest issue with applying extended sampling is that of finding a suitable weight function, and we feel the details of this process warrant special attention as it is fundamental to the success, and timeliness of studies based on extended sampling.

The visited states algorithm (§3.2.1) is simple and effective in many cases, but with challenging systems we can do better. Wang-Landau (§3.2.2) has recently risen to fame for its ability to proactively advance into new regions of phase space; however when applied to the hard problems –and we feel locating (the vicinity of) a single microstate in a high dimensional problem such as this is amongst the most difficult of such problems– Wang-Landau seems lacklustre: the out-of-equilibrium nature leads to sizable errors and thus heavy (and unpredictable) amounts of computation to restore it to equilibrium. By contrast, the transition matrix method appears to deliver the best of both worlds: it operates (essentially) in equilibrium, does not throw existing data away during iteration, can probe the weight function at the extremities of its domain and also utilises rejected Monte Carlo moves. We feel that transition matrix is a clear method of choice for difficult problems and the slightly more involved implementation than other methods is more than compensated for. Successive umbrella sampling (§5.7.2) contrasts with these methods by excelling at 'quick and dirty' estimates of weight functions. Combining any of these methods with windowing (§3.2.4) provides a formidable approach to extended sampling problems. It is worth investing in tools

such as these because of the great generality of extended sampling, and the use of such techniques is sure to grow in the future.

Our first system, the critical point Lennard-Jones fluid (chapter 4), served as a testbed for these extended sampling methods and demonstrates the utility of single simulation comparisons by accessing a reference state and thus estimate the absolute free energy of the system. This was used to verify a known finite size scaling relation. The study also allowed us to develop insight into weight function algorithms in preparation for problems ahead.

The main focus of this thesis, however, was the phase switch Monte Carlo method (§5.4), offering a unique chance to probe fluid–solid coexistence within a theoretically robust and transparent framework. Despite the considerable implementation complexity, once we have constructed a simulation which samples both phases our 'work is done' and, in effect, the particular route taken becomes unimportant. This gives us significant latitude when designing the inter-phase path. By contrast with the earlier lattice switch approach (§5.3), phase switch is more general: any first order phase transition (or other problem involving normally disconnected regions of phase space) can be explored. Although the implementation is complex, the reward is considerable: a general solution for determining phase coexistence properties, and a benchmark against which more approximate methods can be measured.

Our first test case for phase switch was a venerable reference system in statistical mechanics, the hard sphere potential (chapter 5). This allowed us to verify a new implementation of the phase switch method and further explore the relevant extended sampling techniques, this time on a problem with a considerable well depth; and also affording comparison with existing literature values.

Our final study was ambitious: to map the Lennard-Jones freezing line with phase switch Monte Carlo (chapter 6), aiming to provide a results which would set a new benchmark for this well studied reference system. To do so required integrating many threads: we generalised the phase switch method to soft potentials, exploring alternative formulations and novel moves, applied knowledge gained from the earlier exploration of extended sampling, and built on the data analysis framework developed in the context of hard spheres. In all it cannot be said that phase switch is yet routine, with the implementation and execution taking 18 months, consuming 85,000 hours of computer time, and requiring a number of optimisations and developments in order to make it tractable on the available hardware.

A logical next step is to switch from the scaling cutoff selected for the study of the previous chapter (chosen for consistency with existing literature) to a fixed cutoff, which

would be more likely to yield a simple finite size scaling rule.

Directions for future development abound: more realistic potentials, molecular as oppose to atomic systems could be simulated, more extensive studies of polydispersity, and optimisations to the inter-phase path could all form the basis of further work. It would also be attractive to reduce or remove some of the work required to arrive at a phase switch estimate by automating some of the processes involved. A necessary generalisation for some of these ideas is the use of a directional order parameter, or one sensitive to conformational changes to molecules.

More ambitiously, there is no reason why this method will not be scalable to much more realistic potentials, and we speculate that eventually it may be possible to combine phase switch with the quantum Monte Carlo method, significantly extending predictive power of computational simulation. But such developments are still a long way off.

Appendix A

Inverting Metropolis

The typical Metropolis acceptance rule is implemented by accepting an Monte Carlo update if and only if

$$e^{-\beta\Delta\mathcal{H}} > r, \tag{1}$$

where r is a uniform random deviate in $[0, 1]$. To calculate $\Delta\mathcal{H}$ we must consider all interactions including the particles that have moved or changed during the update. However, in hard core simulations as soon as we notice an overlap we know that $\Delta\mathcal{H} = \infty$ and can reject the move, making calculation of the remainder of the interactions redundant. On average, this will more than halve the amount of computation required in the case of rejected moves.

For potentials with highly repulsive, but not hard, cores to their potentials we can still take advantage of this optimisation by 'inverting Metropolis':

$$\Delta\mathcal{H}_{\max} = -k_B T \ln r. \tag{2}$$

The Metropolis criteria can be reformulated to reject a move if $\Delta\mathcal{H} > \Delta\mathcal{H}_{\max}$. This allows us to calculate the boundary energy *before* doing the $\mathcal{O}(N)$ energy calculation. For a purely repulsive potential, we can reject a move as soon as the inequality is satisfied with the incomplete sum of $\Delta\mathcal{H}$ accumulated so far.

For potentials with minima below zero, we can still apply the technique introducing a 'buffer'. Any classical potential is bounded below by some value (-ϵ in Lennard-Jones for example). So if the inequality:

$$\Delta\mathcal{H}' - N\epsilon > \Delta\mathcal{H}_{\max}$$

is violated *at any point* in the energy calculation we can immediately reject the move.

This method was used in the work of chapter 4. It is particularly useful in the grand canonical ensemble, where the chance of creating overlaps is very high.

It was only applied during particle insertion updates, and the cost of making the comparison was amortised over several interactions. In practice, the technique only yielded an increase in speed of around 20 %.

It can be trivially generalised to the extended sampling case *as long as* the order parameter does not depend on the energy (which we have avoided calculating for rejected moves). It was not possible to implement it in the framework of the phase-switch code, since in that case the order parameter $\mathcal{M}$ depends on the energy.

Appendix B

Volume move derivation

Consider a differential phase volume $\mathrm{d}\Gamma$ surrounding a state Γ, of volume V. Now we propose a Monte-Carlo dilation, where the coordinates of all the particles are scaled to lie in the same relative position in a simulation box of a different volume, V':

$$\mathbf{r}_i' = \left(\frac{V'}{V}\right)^{1/3} \mathbf{r}_i \tag{1}$$

This new state is known as Γ', and the move is to a differential phase volume $\mathrm{d}\Gamma'$ surrounding it. Applying detailed balance to moves between these sets of states gives:

$$\mathcal{P}(\Gamma)\mathrm{d}\Gamma\,\pi(\Gamma\to\Gamma')a(\Gamma\to\Gamma') = \mathcal{P}(\Gamma')\mathrm{d}\Gamma'\,\pi(\Gamma'\to\Gamma)a(\Gamma'\to\Gamma), \tag{2}$$

where $\pi(\Gamma \to \Upsilon)$ is the probability of generating a move from Γ to Υ. We arrange this to be symmetric by generating moves thus: if we are in a state with volume V then a destination volume V' is chosen with a uniform random deviate in the interval $[V - \Delta V_{\max}, V + \Delta V_{\max}]$. With this choice the two π factors cancel. The ratio $\mathrm{d}\Gamma/\mathrm{d}\Gamma'$ is in fact $[V/V']^N$ because the projection of $\mathrm{d}\Gamma$ under the dilation includes a scaling for each particle in the system.

Replacing $\mathcal{P}$ from equation 3 on page 67 and rearranging gives:

$$\frac{a(\Gamma\to\Gamma')}{a(\Gamma'\to\Gamma)} = \frac{\delta(\Gamma')}{\delta(\Gamma)}\left[\frac{V'}{V}\right]^N e^{\eta(M')-\eta(M)} e^{-p^*[V'-V]}. \tag{3}$$

The acceptance probability for a move, $a(\Gamma \to \Gamma')$ must lie in the interval $[0, 1]$, but there is still some freedom in choosing how to satisfy this condition (see §1.7). The

choice of Metropolis is typical, yielding:

$$a(\Gamma \rightarrow \Gamma') = \min \left\{ 1,\ \delta(\Gamma') \left[\frac{V'}{V} \right]^N e^{\eta(M')-\eta(M)} e^{-p^*[V'-V]} \right\}, \tag{4}$$

where we assume $\delta(\Gamma) = 1$ since we are moving from that state it must have non-zero probability.

Appendix C

Hard sphere phase–switch acceptance

Consider detailed balance between microstates Γ in one phase and Γ' in the other phase, where they are related by a phase switch move:

$$\mathcal{P}(\Gamma)\mathrm{d}\Gamma\,\pi(\Gamma\rightarrow\Gamma')a(\Gamma\rightarrow\Gamma') = \mathcal{P}(\Gamma')\mathrm{d}\Gamma'\,\pi(\Gamma'\rightarrow\Gamma)a(\Gamma'\rightarrow\Gamma). \tag{1}$$

There is only one possible phase-switch move from any given microstate, so clearly $\pi(\Gamma\rightarrow\Gamma') = \pi(\Gamma'\rightarrow\Gamma)$ and can be cancelled. Rearranging and substituting equation 10 gives:

$$\frac{a(\Gamma\rightarrow\Gamma')}{a(\Gamma'\rightarrow\Gamma)} = e^{\eta_\gamma(0)-\eta_{\gamma'}(0)}\frac{V'}{V}\frac{\mathrm{d}\Gamma'}{\mathrm{d}\Gamma}\frac{\delta(\Gamma')}{\delta(\Gamma)}e^{-p^*[V'-V]}. \tag{2}$$

Noting that since the volume is scaled, but the particle positions are not dilated in a phase-switch, it can be seen that $\mathrm{d}\Gamma'/\mathrm{d}\Gamma = V'/V$. Making the choice of Metropolis for the acceptance probabilities, it is found that:

$$a(\Gamma\rightarrow\Gamma') = \min\left\{1,\, e^{\eta_\gamma(0)-\eta_{\gamma'}(0)}\left[\frac{V'}{V}\right]^2\delta(\Gamma')\,e^{-p^*[V'-V]}\right\}, \tag{3}$$

where we have taken $\delta(\Gamma) = 1$ since we can always assume the simulation is moving from a physically realisable microstate.

Appendix D

Lennard-Jones phase–switch acceptance

Consider a differential phase volume $d\Gamma$ surrounding a state Γ, of volume V. Now we propose a Monte-Carlo dilation, where the coordinates of all the particles are scaled to lie in the same relative position in a simulation box of a different volume, V'. This new state is known as Γ', and the move is to a differential phase volume $d\Gamma'$ surrounding it. Applying detailed balance to moves between these sets of states gives:

$$\mathcal{P}(\Gamma)d\Gamma\,\pi(\Gamma\rightarrow\Gamma')a(\Gamma\rightarrow\Gamma') = \mathcal{P}(\Gamma')d\Gamma'\,\pi(\Gamma'\rightarrow\Gamma)a(\Gamma'\rightarrow\Gamma), \tag{1}$$

where $\pi(\Gamma \rightarrow \Gamma')$ is the probability of generating a move from Γ to Γ'. We arrange this to be symmetric by generating moves thus: if we are in a state with volume V then a destination volume V' is chosen with a uniform random deviate in the interval $[V - \Delta V_{\max}, V + \Delta V_{\max}]$. With this choice the two π factors cancel. The ratio $d\Gamma/d\Gamma'$ is in fact $[V/V']^N$ because the projection of $d\Gamma$ under the dilation includes a scaling for each particle in the system.

Replacing $\mathcal{P}$ from equation 4 on page 90 and rearranging gives:

$$\frac{a(\Gamma\rightarrow\Gamma')}{a(\Gamma'\rightarrow\Gamma)} = \frac{\delta(\Gamma')}{\delta(\Gamma)}\left[\frac{V'}{V}\right]^{N+1}\frac{\eta(\Gamma)}{\eta(\Gamma')}e^{-\beta P[V'-V]}. \tag{2}$$

The acceptance probability for a move, $a(\Gamma \rightarrow \Gamma')$ must lie in the interval $[0, 1]$, but there is still some freedom in choosing how to satisfy this condition. The choice of

Metropolis is typical, yielding:

$$a(\Gamma \rightarrow \Gamma') = \min \left\{ 1, \left[\frac{V'}{V} \right]^{N+1} e^{-\eta(\Gamma)-\eta(\Gamma')} e^{-\beta P[V'-V]} \right\}. \tag{3}$$

References

[1] O. Mishima and H. E. Stanley, *Nature* **396**, 329 (1998).

[2] L. Glasser, *J. Chem. Educ.* **81**, 414 (2004).

[3] M. Plischke and B. Bergersen, "Equilibrium Statistical Mechanics", *World Scientific*, 2nd edition (1994).

[4] C. Domb and M. S. Green, "Phase Transitions and Critical Phenomena", *Academic Press* (1975).

[5] J. Binney et al, "The Theory of Critical Phenomena: An introduction to the Renormalisation Group", *Oxford University Press* (1992).

[6] D. Frenkel and B. Smit, "Understanding Molecular Simulation", *Academic Press*, 2nd edition, ch. 4 (2002).

[7] W. M. C. Foulkes, L. Mitas, R. J. Needs, and G. Rajagopal, *Rev. Mod. Phys.* **73**, 33 (2001).

[8] F. Mandl, "Statistical Mechanics", *Wiley* (1971).

[9] R. P. Feynman, "Statistical Mechanics", *Addison-Wiley* (1972).

[10] R. P. Feynman, "Statistical Mechanics", *Addison-Wiley*, ch. 1 (1972).

[11] D. Frenkel and B. Smit, "Understanding Molecular Simulation", *Academic Press*, 2nd edition (2002).

[12] L. Verlet, *Phys. Rev.* **159**, 98 (1967).

[13] W. C. Swope, H. C. Andersen, P. H. Berens, and K. R. Wilson, *J. Chem. Phys.* **76**, 637 (1982).

[14] J. B. Marion and S. T. Thornton, "Classical Dynamics of Particles and Systems.", *Harcourt Brace Jovanovich*, 3rd edition (1988).

[15] N. Metropolis and S. Ulam, *J. Am. Stat. Assoc.* **44**, 247, 335 (1949)

[16] W. K. Hastings, *Biometrika* **57**, 97 (1970).

[17] M. E. J. Newman and G. T. Barkema, "Monte Carlo methods in statistical physics", *Clarendon Press*, Oxford, 2nd edition, 36 (2001).

[18] J. von Neumann, *Nat. Bureau Standards*, 12, 36 (1951).

[19] D. Frenkel and B. Smit, "Understanding Molecular Simulation", *Academic Press*, 2nd edition, ch. 3 (2002).

[20] A. Milchev, K. Binder and D. W. Heerman, *Z. Phys. B* **63**, 521 (1986).

[21] Kirkpatrick and Stoll, *J. ComPhys.* **40**, 517 (1981).

[22] N. Zierler, *Information and Control* **15**, 67 (1969).

[23] A. M. Ferrenberg, D. P. Landau and Y. J. Wong, *Phys. Rev. Lett.* **69**, 3382 (1992).

[24] A. Heuer, B. D unweg and A. M. Ferrenberg, *ComPhys. Comm.* **103**, 1, 1 (1997).

[25] D. Knuth, "The Art of Computer Programming, Volume 2: Seminumerical Algorithms", *Addison-Wiley*, 3rd edition, ch. 3 (1997).

[26] W. H. Press et al, "Numerical Recipes in C", *Cambridge University Press*, 2nd edition, ch. 7 (1992).

[27] J. Liu and E. Luijten, *Phys. Rev. E* **71**, 6 (2005).

[28] D. R. Herschbach, H. S. Johnston, and D. Rapp, *J. Chem. Phys.* **31**, 1652 (1959).

[29] W. G. Hoover and F. H. Ree, *J. Chem. Phys.* **47**, 4873 (1969).

[30] D. Frenkel and A. J. C. Ladd, *J. Chem. Phys.* **81**, 7 (1984).

[31] A. M. Ferrenberg and R. H. Swendsen, *Phys. Rev. Lett.* **61**, 23, 2635 (1988).

[32] A. M. Ferrenberg and R. H. Swendsen, *Phys. Rev. Lett.* **63**, 12, 1195 (1989).

[33] A. M. Ferrenberg, D. P. Landau and R. H. Swendsen, *Phys. Rev. E* **51**, 5092 (1995).

[34] G. M. Torrie and J. P. Valleau, *Chem. Phys. Lett.* **28**, 578 (1974).

[35] G. R. Smith and A. D. Bruce, *J. Phys. A: Math. Gen.* **28**, 23, 6623 (1995).

[36] J. P. Valleau and D. N. Card, *J. Chem. Phys.* **57**, 12, 5457 (1972).

[37] G. M. Torrie and J. P. Valleau. *J. ComPhys.* **23**, 187 (1977).

[38] C. Jarzinsky, *Phys. Rev. Lett.* **87**, 2690 (1997).

[39] D. Frenkel and A. J. C. Ladd, *J. Chem. Phys.* **81**, 3188 (1984).

[40] J. M. Polson, E. Trizac, S. Pronk and D. Frenkel, *J. Chem. Phys.* **112**, 5339 (2000).

[41] D. A. Kofke, *Mol. Phys.* **78**, 1331 (1993).

[42] D. A. Kofke, *J. Chem. Phys.* **98**, 4149 (1993).

[43] R. Agrawal and D. A. Kofke, *Mol. Phys.* **85**, 43 (1995).

[44] R. Agrawal and D. A. Kofke, *Phys. Rev. Lett.* **74**, 122 (1995)

[45] D. A. Kofke and P. G. Bolhuis, *Phys. Rev. E* **59**, 618 (1999).

[46] F. A. Escobedo and J. J. de Pablo *J. Chem. Phys.* **106**, 2911 (1997).

[47] A. Z. Panagiotopoulos, *Mol. Phys.* **61**, 813 (1987).

[48] A. Z. Panagiotopoulos, N. Quirke, M. Stapleton and D. J. Tildesley, *Mol. Phys.* **63**, 527 (1988).

[49] F. Escobedo, *J. Chem. Phys.* **110**, 11999 (1999).

[50] R. Shetty and F. A. Escobedo, *J. Chem. Phys* **116**, 18 (2002)

[51] E. Marinari and G. Parisi, *Europhys. Lett.* **19**, 451 (1992).

[52] C. J. Geyer and E. A. Thompson, *J. Am. Stat. Assoc.* **90**, 909 (1995).

[53] M. C. Tesi, E. J. Janse van Rensburg, E. Orlandini and S. G. Whittington, *J. Stat. Phys.* **82**, 155 (1996).

[54] M. Falcioni and M. W. Deem, *J. Chem. Phys.* **110**, 1754 (1999).

[55] Q. Yan and J. J. de Pablo, *J. Chem. Phys.* **111**, 9509 (1999).

[56] G. Grochla, *J. Chem. Phys.* **120**, 2122 (2004).

[57] D. M. Elke, J. F. Brennecke and E. J. Maginn, *J. Chem. Phys.* **122**, 014115 (2005).

[58] G. Grochla, *J. Chem. Phys.* **122**, 046101 (2005).

[59] E. A. Mastny and J. J. de Pablo, *J. Chem. Phys.* **122**, 124109 (2005).

[60] E. A. Mastny, personal communication (2005).

[61] F. A. Escobedo, *J. Chem. Phys.* **123**, 044110 (2005).

[62] A. Laio and M. Parrinello, *Proc. Nat. Acad. Sci. USA* **99**, 20, 12562 (2002).

[63] A. R. Acharya, "Free energy differences: Representations, estimators and sampling strategies", PhD Thesis, *University of Edinburgh* (2004).

[64] Fugao Wang and D. P. Landau, *Phys. Rev. E*, **64** (056101), 2001

[65] S. Trebst, D. A. Huse, and M. Troyer, *Phys. Rev. E* **70**, 046701 (2004).

[66] B. A. Berg and T. Neuhaus, *Phys. Rev. Lett.* **68**, 1, 9 (1992).

[67] J. Lee, *Phys. Rev. Lett.* **71**, 2, 211 (1993).

[68] Fugao Wang and D. P. Landau, *Phys. Rev. Lett.* **86**, 2050 (2001).

[69] M. S. Shell et al, *Phys. Rev. E* **66**, 056703 (2002).

[70] G. R. Smith and A. D. Bruce, *J. Phys. A* **28**, 6623 (1995).

[71] J. S. Wang et al, *Phys. Rev. Lett.* **82**, 476 (1999).

[72] J. S. Wang and R. H. Swendsen, *J. Stat. Phys.* **106**, 245 (2002).

[73] J. R. Errington, *J. Chem. Phys.* **120**, 3130 (2004).

[74] M. Fitzgerald, R. R. Picard and R. N. Silver, *J. Stat. Phys.* **98** 1, 321 (2004).

[75] D. Ceperley, G. V. Chester and M. H. Kalos, *Phys. Rev. B* **16**, 7, 3081 (1977).

[76] D. Frenkel, *Proc. Nat. Acad. Sci. USA* **101**, 51, 17571 (2004).

[77] J. G. Powles, *Physica A* **126**, 1, 289 (1984).

[78] J. M. Caillol, *J. Chem. Phys* **109**, 12, 4885 (1998).

[79] J. Pérez-Pellitero et al, *J. Chem. Phys* **125**, 5, (2006).

[80] N. B. Wilding, *Phys. Rev. E*, **52**, 602 (1995).

[81] A. D. Bruce, *J. Phys. A: Math. Gen.* **28**, 3345 (1995).

[82] N. B. Wilding, *Am. J. Phys.*, **69**, 11 (2001).

[83] D. Frenkel and B. Smit, "Understanding Molecular Simulation", *Academic Press*, 2nd edition, 113 (2002).

[84] A. D. Bruce and N. B. Wilding, *Phys. Rev. E*, **60** 3748 (1999).

[85] A. D. Bruce and N. B. Wilding, *Phys. Rev. Lett.* **79**, 3002 (1997).

[86] D. Frenkel and B. Smit, "Understanding Molecular Simulation", *Academic Press*, 2nd edition, 104 (2002).

[87] P. Virnau and M. M uller, *J. Chem. Phys.* **120**, 23, 10925 (2004).

[88] C. Zhou and R. N. Bhatt, *Phys. Rev. E* **72**, 025701 (2005).

[89] M. E. J. Newman and G. T. Barkema, "Monte Carlo Methods in Statistical Physics", *Clarendon Press*, 31 (1999).

[90] A. D. Bruce et al, *Phys. Rev. E* **61**, 1 (2000).

[91] D. Frenkel and A. J. C. Ladd, *J. Chem. Phys.* **81**, 3188 (1984).

[92] J. R. Errington, *J. Chem. Phys.* **118**, 9915 (2003).

[93] J. R. Errington, *Phys. Rev. E* **67**, 012102 (2003).

[94] W. G. Hoover and F. H. Ree, *J. Chem. Phys.* **47**, 4873 (1967).

[95] W. G. Hoover and F. H. Ree, *J. Chem. Phys.* **49**, 3609 (1968).

[96] W. G. Hoover, S. G. Gray, and K. W. Johnson, *J. Chem. Phys* **55**, 3, 1128 (1971).

[97] D. Frenkel and A. J. C. Ladd, *J. Chem. Phys.* **81**, 3188 (1984).

[98] E. R. Cowley and J. A. Barker, *J. Chem. Phys* **73**, 7, 3452 (1980).

[99] A. R. Denton and N. W. Ashcroft, *Phys. Rev. A* **39**, 4701 (1989).

[100] J. F. Lutsko and M. Baus, *Phys. Rev. A* **41**, 6647 (1990).

[101] W. K. Kegel, H. Reiss, and H. N. W. Lekkerkerker, *Phys. Rev. Lett.* **83**, 25 (1999).

[102] P. N. Pusey and W. van Megen, *Nature (London)* **320**, 340 (1986).

[103] A. N. Jackson, A. D. Bruce, and G. J. Ackland, *Phys. Rev. E* **65**, 036710 (2002).

[104] N. B. Wilding and A. D. Bruce, *Phys. Rev. Lett.* **85**, 5138 (2000).

[105] N. B. Wilding, *ComPhys. Comm.* **146**, 99 (2002).

[106] N. B. Wilding, *ComPhys. Com.* **146**, 99 (2002).

[107] D. Frenkel and B. Smit, "Understanding Molecular Simulation", *Academic Press*, 2nd edition, ch. 2 (2002).

[108] J. D. Honeycutt and H. C. Andersen, *J. Phys. Chem* **91**, 4950 (1987).

[109] J. R. Morris and X. Song, *J. Chem. Phys.* **116**, 9352 (2002).

[110] G. C. McNeil-Watson and N. B. Wilding, *J. Chem. Phys* **124**, 064504 (2006).

[111] W. H. Press et al, "Numerical Recipes in C", *Cambridge University Press*, Second Edition, `dzbrent` function (1992).

[112] M. H. Quenouille, *J. Royal Stat. Soc. B* **11**, 18 (1949).

[113] M. H. Quenouille, *Biometrika* **61**, 353 (1956).

[114] B. Efron and R. J. Tibshirani, "An Introduction to the Bootstrap", *Chapman and Hall*, London (1993).

[115] M. C. K. Yang and David H. Robinson, "Understanding and Learning Statistics by Computer", *World Scientific*, Singapore (1986).

[116] M. E. J. Newman and G. T. Barkema, "Monte Carlo Methods in Statistical Physics", *Clarendon Press*, 211 (1999).

[117] P. A. Monson and D. A. Kofke, *Adv. Chem. Phys.* **115**, 113 (2000).

[118] A.D. Bruce and N.B. Wilding, Adv. Chem. Phys. **127**, 1 (2003).

[119] M. A. Barroso and A. L. Ferreira, *J. Chem. Phys.* **116**, 7145 (2002).

[120] N. B. Wilding and A. D. Bruce, *Phys. Rev. Lett.* **85**, 5138 (2000).

[121] A. M. Ferrenberg and R. H. Swendsen, *Phys. Rev. Lett.* **61**, 2635 (1988).

[122] G. R. Smith and A. D. Bruce, *J. Phys. A* **28**, 6623 (1995).

[123] M. Fitzgerald, R. R. Picard, R. N. Silver, *Europhys. Lett.* **46**, 282 (1999).

[124] M. Fitzgerald, R. R. Picard, R. N. Silver, *J. Stat. Phys.* **98**, 321 (2000).

[125] M. P. Allen and D. J. Tildesley, "Computer Simulation of Liquids", *Oxford Press* (1988).